

# CA Process Automation

## Guia do Criador de Conteúdo

Release 04.2.00



A presente documentação, que inclui os sistemas de ajuda incorporados e os materiais distribuídos eletronicamente (doravante denominada Documentação), destina-se apenas a fins informativos e está sujeita a alterações ou remoção por parte da CA a qualquer momento. Esta Documentação contém informações proprietárias da CA e não pode ser copiada, transferida, reproduzida, divulgada, modificada nem duplicada, parcial ou completamente, sem o prévio consentimento por escrito da CA.

Se o Cliente for um usuário licenciado do(s) produto(s) de software referido(s) na Documentação, é permitido que ele imprima ou, de outro modo, disponibilize uma quantidade razoável de cópias da Documentação para uso interno seu e de seus funcionários envolvidos com o software em questão, contanto que todos os avisos de direitos autorais e legendas da CA estejam presentes em cada cópia reproduzida.

O direito à impressão ou, de outro modo, à disponibilidade de cópias da Documentação está limitado ao período em que a licença aplicável ao referido software permanecer em pleno vigor e efeito. Em caso de término da licença, por qualquer motivo, fica o usuário responsável por garantir à CA, por escrito, que todas as cópias, parciais ou integrais, da Documentação sejam devolvidas à CA ou destruídas.

NA MEDIDA EM QUE PERMITIDO PELA LEI APLICÁVEL, A CA FORNECE ESTA DOCUMENTAÇÃO "NO ESTADO EM QUE SE ENCONTRA", SEM NENHUM TIPO DE GARANTIA, INCLUINDO, ENTRE OUTROS, QUAISQUER GARANTIAS IMPLÍCITAS DE COMERCIALIZABILIDADE, ADEQUAÇÃO A UM DETERMINADO FIM OU NÃO VIOLAÇÃO. EM NENHUMA OCASIÃO, A CA SERÁ RESPONSÁVEL PERANTE O USUÁRIO OU TERCEIROS POR QUAISQUER PERDAS OU DANOS, DIRETOS OU INDIRETOS, RESULTANTES DO USO DA DOCUMENTAÇÃO, INCLUINDO, ENTRE OUTROS, LUCROS CESSANTES, PERDA DE INVESTIMENTO, INTERRUPTÃO DOS NEGÓCIOS, FUNDO DE COMÉRCIO OU PERDA DE DADOS, MESMO QUE A CA TENHA SIDO EXPRESSAMENTE ADVERTIDA SOBRE A POSSIBILIDADE DE TAIS PERDAS E DANOS.

O uso de qualquer software mencionado na Documentação é regido pelo contrato de licença aplicável, e tal contrato não deve ser modificado de nenhum modo pelos termos deste aviso.

O fabricante desta Documentação é a CA.

Fornecida com "Direitos restritos". O uso, duplicação ou divulgação pelo governo dos Estados Unidos está sujeita às restrições descritas no FAR, seções 12.212, 52.227-14 e 52.227-19(c)(1) - (2) e DFARS, seção 252.227-7014(b)(3), conforme aplicável, ou sucessores.

Copyright © 2010 CA. Todos os direitos reservados. Todas as marcas comerciais, nomes de marcas, marcas de serviço e logotipos aqui mencionados pertencem às suas respectivas empresas.

## Referências a produtos da CA Technologies

O documento faz parte de uma biblioteca com referências aos seguintes produtos CA Technologies:

- CA Catalyst para o CA Service Desk Manager (CA Catalyst Conector do CA SDM)
- CA Client Automation (anteriormente CA IT Client Manager)
- CA Configuration Automation (anteriormente, CA Cohesion® Application Configuration Manager)
- CA CMDB
- CA eHealth®
- CA EEM (Embedded Entitlements Manager)
- CA Infrastructure Insight (anteriormente Bundle: CA Spectrum IM & CA NetQoS Reporter Analyzer combinados)
- CA NSM
- CA Process Automation (anteriormente, CA IT Process Automation Manager)
- CA Service Catalog
- CA SDM (CA Service Desk Manager)
- CA Service Operations Insight (CA SOI) (anteriormente CA Spectrum® Service Assurance)
- CA SiteMinder®
- CA Workload Automation AE

## Entrar em contato com o Suporte técnico

Para assistência técnica online e uma lista completa dos locais, principais horários de atendimento e números de telefone, entre em contato com o Suporte técnico pelo endereço <http://www.ca.com/worldwide>.



# Índice

---

## Capítulo 1: Introdução ao Process Automation 11

Fases do desenvolvimento de processos .....	13
Fase I: Análise de requisitos .....	14
Identificar processos para automatizar .....	15
Identificar onde os processos são executados.....	15
Identificar as etapas dos processos .....	16
Identificar as interdependências.....	16
Identificar as dependências externas .....	17
Identificar restrições de tempo de execução.....	17
Fase II: Design e implementação.....	17
Identificar operadores e objetos de processos.....	18
Configurar os operadores e as etapas.....	21
Otimizar para reutilização da modularidade e de componentes.....	23
Definir o início e o monitoramento do processo .....	24
Definir alertas.....	28
Fase III: Teste e implantação .....	28
Testar o processo com objetos relacionados.....	29
Gerar a documentação do processo .....	30
Definir a versão de release do processo e de objetos relacionados .....	30
Montar o processo com objetos relacionados.....	30
Implantar uma versão da release do processo com objetos relacionados.....	31
Especificar instruções de importação .....	31

## Capítulo 2: Guia de Introdução 33

Vá até o CA Process Automation e efetue logon. ....	34
Definir as configurações do usuário .....	35
A interface do usuário do CA Process Automation .....	36
Páginas principais do aplicativo .....	38
Controles comuns de interface do usuário .....	53
Explorar o conteúdo pronto para uso .....	55
Usar o CA EEM para alterar sua senha do CA Process Automation .....	56
Navegadores da web.....	57

## Capítulo 3: Navegador da biblioteca 59

Personalizar o Navegador da biblioteca.....	59
Pesquisar o Navegador da biblioteca .....	59

---

Pesquisar por informações da versão da release .....	61
Pesquisar por informações da versão .....	62
Pesquisar por informações de trilha de auditoria .....	63
Trabalhando com objetos e pastas .....	64
Tipos de objetos de automação .....	64
Criar uma pasta .....	66
Criar um objeto .....	67
Editar um objeto .....	68
Versões .....	79
Excluir ou restaurar um objeto ou uma pasta .....	91
Copiar e colar um objeto ou uma pasta .....	92
Recortar e colar um objeto ou uma pasta .....	93
Como trabalhar com conteúdo não modificável .....	94

## **Capítulo 4: Criando processos** **97**

Criador de processos .....	98
Operadores e links: os blocos de criação .....	100
Criar um objeto de processo .....	101
Criar um processo .....	102
Operadores de processo .....	104
Operador Iniciar .....	105
Adicione os operadores a um processo .....	106
Operadores lógicos .....	107
Operador Interromper: êxito ou falha .....	109
Portas e links do operador de processo .....	110
Adicionar portas e links do operador .....	112
Portas de saída e expressões personalizadas .....	113
Quebrar um link para fins de legibilidade .....	115
Loops e iterações de processo .....	115
Variáveis de sistema para loops .....	116
Loop de um operador em um processo .....	117
Loop de uma série de operadores .....	121
Loop de um processo .....	124
Controle de processo .....	126
Processos filho .....	127
Processo embutido .....	132
Rotas de processo .....	134
Criar pistas horizontais ou verticais .....	135
Gerenciar raias .....	136
Regras de manipulação de rotas .....	137
Versões de processos .....	138

---

Documentar um processo .....	138
Adicionar comentários a um processo .....	139
Defina o nome de um operador em um processo .....	139
Alterar e exibir informações do operador em um processo .....	140
Conteúdo Self-Contained .....	140
Links de conteúdo independente.....	141
Navegar até uma parte específica de um processo .....	143
Multilocação e CA Process Automation .....	144
Criar um processo existente ciente dos multilocação .....	145
Herdar segurança em subprocessos .....	146
Adicionar variáveis na hora da inicialização.....	146
Processos de multilocação usando a exibição de processos .....	146
O editor de código do CA Process Automation .....	146

## **Capítulo 5: Operadores e ícones** **153**

Operadores.....	154
Configurar propriedades do operador .....	156
Recuperação automática .....	157
Java e JARs externos.....	158
Operadores personalizados .....	164
Seus operadores favoritos .....	190
Conectores .....	191
Ícones dos operadores .....	192
Ícones de status do operador .....	193
Criando, editando e aplicando ícones personalizados .....	194

## **Capítulo 6: Conjuntos de dados e parâmetros** **203**

Conjuntos de dados.....	203
Criar um objeto de conjunto de dados nomeado .....	205
Definir páginas e variáveis de conjuntos de dados .....	207
Modificar um conjunto de dados .....	213
Exibir ou copiar uma expressão do conjunto de dados .....	214
Ler valores do sistema operacional em variáveis do conjunto de dados.....	214
Parâmetros do processo .....	223
Propriedades do operador .....	223
Parâmetros calculados .....	238

## **Capítulo 7: Formulários** **275**

Formulários de solicitação inicial .....	275
Monitorar instâncias de formulário de solicitação inicial e instâncias de processo .....	276

---

Formulários de solicitação de interação .....	277
Criador de formulários .....	279
Elementos do formulário .....	282
Propriedades do elemento do formulário .....	284
Eventos de elemento do formulário .....	297
Funções de elemento do formulário .....	300
Inicializar variáveis do formulário .....	367

## **Capítulo 8: Recursos 369**

Como os recursos funcionam .....	370
Criar um objeto Recurso .....	371
Edite um objeto Recurso .....	372
Monitorar e editar recursos .....	374
Adicionar um operador Gerenciar recursos em um processo .....	375
Definir ações de recursos .....	376
Verificar e responder a recursos indisponíveis .....	377
Especifique um intervalo de tempo limite .....	378
Especificar as configurações de Ação e de Disponibilidade de recursos .....	379
Verificar a disponibilidade do recurso sem executar ações .....	380

## **Capítulo 9: Calendários, programações, tarefas e disparadores 381**

Calendários .....	381
Criar um objeto Calendário .....	382
Criador de calendários básico .....	383
O Criador de calendários avançado .....	384
Criador de calendários: guia Visualizar .....	394
Excluir calendários .....	396
Programações .....	397
Criar um objeto de programação .....	400
Programar processos e tarefas do operador .....	401
Visualizar todas as ocorrências de uma tarefa programada .....	403
Usando programações .....	404
Monitorar programações ativas .....	405
Monitorar todas as ocorrências de todas as tarefas programadas .....	405
Gerenciamento de tarefas .....	407
Atribuir uma tarefa a um usuário .....	407
Lista de tarefas .....	409
Administrar disparadores .....	410
Controle de processos por meio de um aplicativo externo com chamadas SOAP .....	411
Como os disparadores de arquivos e emails funcionam .....	411
Considerações sobre a entrada da SNMP trap .....	415

---

## Capítulo 10: Executando, testando e depurando processos 419

A página Operações .....	421
Filtros para instâncias do processo .....	422
Filtrar objetos exibidos por um atalho .....	423
Objetos Exibição de processos .....	424
Objetos de pacote de conteúdo .....	425
Regras de execução .....	427
Segurança de tempo de execução .....	428
As propriedades afetam a segurança dos processos em execução .....	428
Diretrizes para definir a Segurança de tempo de execução para um processo .....	430
Tratamento de exceção .....	431
Criar manipuladores de exceção .....	432
Executar processos de forma interativa .....	435
Iniciar um processo da biblioteca .....	436
Iniciar um processo da biblioteca como Suspenso .....	437
Iniciar um processo durante a edição .....	438
Abrir uma instância de um processo .....	438
Estados do processo .....	439
Depurar um processo .....	439
Suspender um processo .....	439
Alterar se processos são descarregados na conclusão .....	440
Definir e remover pontos de interrupção de um processo .....	440
Depurar um processo Java .....	442
Redefinir um processo .....	443
Cancelar um processo .....	443
Controlar a ramificação de um processo .....	443
Desativar operadores ou desativar ramificações .....	444
Cancelar um operador .....	444
Redefinir os operadores em um processo .....	445
Reiniciar a execução de um processo suspenso .....	445
Simular o processamento dos operadores .....	446
Simular o processamento de um operador selecionado .....	447
Simular o processamento de um processo inteiro .....	448

## Capítulo 11: Gerando e usando a documentação do processo 449

Sobre a documentação do processo .....	449
Gerar documentação do processo .....	450

## Capítulo 12: Liberar objetos para outro ambiente 453

Versões da release .....	454
--------------------------	-----

---

Exibir informações da versão de release.....	454
Definir a versão da release de objetos a serem exportados .....	455
Definir a versão base da versão da release dos objetos .....	457
Definir a versão da release do pacote de conteúdo como atual .....	458
Exportando e importando .....	458
Exportando.....	459
Importando .....	474
Valores mantidos na versão inicial de um objeto copiado .....	483

## **Apêndice A: Formatar especificadores para datas** **485**

## **Apêndice B: Usando máscaras para especificar padrões nas sequências de caracteres** **489**

Sintaxe da máscara.....	489
Máscara de exemplo .....	493

## **Apêndice C: Como os destinos de um operador são processados** **495**

Como os destinos para um operador podem ser especificados .....	495
Processando um destino especificado como um endereço IP ou FQDN .....	496
Processando um destino especificado como a ID de um agente ou orquestrador.....	501
Caso de Uso: rastrear processos recuperados por meio dos logs.....	502
Exemplo da Recuperação automática dos Operadores .....	503

# Capítulo 1: Introdução ao Process Automation

---

As equipes de tecnologia da informação como as suas estão automatizando os respectivos processos de TI. Com a definição, automação e orquestração de processos nos sistemas, você pode melhorar a produtividade com a aplicação de padrões entre os departamentos. A automação ajuda a sua organização a:

- Reduzir despesas operacionais.
- Aumentar a produtividade da equipe.
- Agilizar a entrega de serviços de TI.
- Melhorar a qualidade do serviço.
- Aplicar políticas de conformidade

Automatize os processos operacionais que, de outra forma, poderiam ser manuais, demorados, inconsistentes e sujeitos a erro. É possível automatizar os processos de TI distribuídos em diversos sistemas de várias organizações.

## Bem-vindo ao CA Process Automation



Use o CA Process Automation para criar, testar, gerenciar e relatar processos automatizados que oferecem suporte às operações de TI e aos ambientes de produção. O CA Process Automation agiliza a entrega de serviços de TI, reduzindo, ao mesmo tempo, os erros manuais. Você pode:

Use o CA Process Automation no seu navegador web para gerenciar, criar e implantar processos em uma interface fácil de usar que oferece suporte aos seguintes recursos essenciais:

- Autoria visual, incluindo criações baseadas em ícone arrastar e soltar conhecidas.
- Monitoramento de processo corporativo com a capacidade de pausar, modificar e retomar os processos em execução ativos.
- Exibições baseada em funções com opções de layout flexíveis.
- Objetos amigáveis aos usuários. Processos, calendários, programações, conjuntos de dados, formulários e outros objetos de automação são representados como objetos que podem ser clicados.
- Fácil organização de objetos de automação por meio de uma biblioteca de pastas hierárquicas.
- Controle de versão de objeto, incluindo disponibilização, reserva e seleção de versões atuais e de linha de base.
- Recursos de assistência ativos para ajudá-lo a trabalhar com conjuntos de dados, variáveis e expressões.
- Operadores de processo que se integram a outros aplicativos e sistemas.

Esta seção contém os seguintes tópicos:

[Fases do desenvolvimento de processos](#) (na página 13)

[Fase I: Análise de requisitos](#) (na página 14)

[Fase II: Design e implementação](#) (na página 17)

[Fase III: Teste e implantação](#) (na página 28)

## Fases do desenvolvimento de processos

Antes de trabalhar com o CA Process Automation, é importante que você, como um desenvolvedor de conteúdo ou designer, esteja familiarizado com os conceitos por trás do desenvolvimento do processo. O restante desta seção fornece detalhes das fases e etapas envolvidas na automatização de processos de TI.



O desenvolvimento do processo inclui as seguintes fases:

### **Fase I: Análise de requisitos (na página 14)**

Identificar e caracterizar os processos de automação necessários em sua organização.

### **Fase II: Design e implementação (na página 17)**

Mapear os requisitos para os objetos de automação e configurar os elementos. Identificar os elementos existentes para reutilização, os módulos necessários de terceiros (por exemplo, processos relacionados a bancos de dados ou dados de administradores de banco de dados) e os módulos que você deve desenvolver sozinho. É possível usar as ferramentas de desenvolvimento do CA Process Automation para importar ou definir esses objetos de automação em sua biblioteca.

### **Fase III: Teste e implantação (na página 28)**

Exportar todos os dados de automação relevantes para um arquivo e facilitar sua importação para o ambiente de produção do CA Process Automation. Fornecer instruções aos administradores de produção explicando como configurar os elementos para funcionar no ambiente de produção.

## Fase I: Análise de requisitos

Essa fase concentra-se nas informações que devem ser coletadas e os problemas que devem ser solucionados antes de você criar um pacote ou um processo de gerenciamento do CA Process Automation. Os sites e as soluções de gerenciamento específicas podem exigir tarefas adicionais.

Inclua as seguintes tarefas em uma típica análise de requisitos:

- Identificar os processos a serem automatizados.
- Identificar onde os processos são executados.
- Identificar as etapas dos processos e descrever os resultados de cada etapa.
- Identificar as interdependências.
- Identificar as dependências externas.
- Identificar as restrições de tempo de execução.

Os bons candidatos à automação de processos atendem a um ou mais dos seguintes requisitos de início:

- Você pode identificar várias tarefas, geralmente, interdependentes.
- Você pode identificar os requisitos de sobreposição de recursos.
- O processo pode ser executado na rede ou em plataformas diferentes.
- Você pode aplicar restrições de data e hora para programar o processo.
- Outros processos, aplicativos ou usuários podem disparar o processo.
- Administradores de sistema ou de banco de dados, operadores ou outros usuários podem executar o processo sob demanda.

## Identificar processos para automatizar

A primeira etapa no processo de desenvolvimento do CA Process Automation é a de identificar processos para automação. Um processo geralmente consiste em vários subprocessos.

### Siga estas etapas:

1. Identifique a tarefa principal ou o caso de uso, como *mover dados de bancos de dados transacionais distribuídos para um único repositório de dados*.
2. Identifique os subprocessos para tarefas subordinadas, como:
  - a. *Extrair dados de origens diferentes*
  - b. *Transformar os dados extraídos*
  - c. *Carregar os dados transformados*
3. Separe os componentes que são comuns em vários processos em subprocessos modulares. Semelhante à reutilização de código, a modularização de subprocessos compartilhada poupa tempo de desenvolvimento e manutenção.

**Observação:** mais adiante neste guia, você aprenderá que os subprocessos correspondem a [objetos](#) (na página 64) de processo separados no Navegador da biblioteca.

## Identificar onde os processos são executados.

Identificar o local em que você deseja executar os seguintes componentes:

- O processo propriamente dito
- Subprocessos
- Tarefas do processo e subprocessos

**Observação:** mais adiante neste guia, você aprenderá o que esses locais correspondem a *touchpoints* que executam processos ou etapas individuais nos processos. Seu administrador configura os touchpoints.

## Identificar as etapas dos processos

Liste cada etapa executada como parte do processo ou subprocesso.

### Siga estas etapas:

1. Identifique cada etapa. Os exemplos incluem:
  - Execução de um aplicativo
  - Obtenção de dados de uma origem remota.
  - Execução de um ou mais cálculos.
  - Solicitação ao usuário de informações com um formulário
  - Início de um subprocesso.
2. Em cada etapa, determine os possíveis resultados a seguir:
  - a. Resultados normais
  - b. Resultados anormais
  - c. Qualquer resultado condicional

Uma etapa pode gerar resultados condicionais. Quando o resultado é maior do que o valor especificado, a etapa gera um determinado resultado. Quando o resultado é menor do que o valor especificado, a etapa gera um resultado diferente. Vários resultados calculados determinam os caminhos subsequentes pelas ramificações de um processo.

**Observação:** mais adiante neste guia, você aprenderá que as etapas na criação de um processo correspondem aos *operadores* (na página 100).

## Identificar as interdependências

As interdependências determinam o fluxo lógico através de um processo.

### Siga estas etapas:

1. Determine quais etapas de um processo devem ser anteriores ou posteriores às outras etapas.
2. Determine como o resultado de cada etapa afeta as etapas posteriores.
3. Para uma etapa com mais de um resultado, identifique cada resultado e liste a sequência de etapas separada que deve ser executada após cada resultado.

**Observação:** o aplicativo executa cada resultado separadamente. Por exemplo, um resultado anormal de uma etapa pode fazer com que um processo para alertar outra etapa aguarde uma entrada. Para a mesma etapa ou o mesmo operador, um resultado personalizado pode ativar uma ramificação separada do processo para processar uma condição de erro.

## Identificar as dependências externas

Identificar as dependências externas das etapas em um processo, como:

- Aplicativos e sistemas remotos.
- Recursos compartilhados, como sistemas de arquivos ou bancos de dados.
- Processos sincronizados e outros processos que são executados paralelamente.

## Identificar restrições de tempo de execução

Identificar os recursos necessários a um processo em tempo de execução. Por serem compartilhados, os recursos sempre têm fornecimento limitado; por exemplo, processadores, memória ou acesso a outros arquivos, bancos de dados, ou hardware. Esteja ciente destes dois tipos principais de restrições de tempo de execução:

- As **restrições de tempo** definem quando um processo deve ser concluído. Os exemplos incluem backups executados à noite ou as atividades de manutenção do sistema que são executadas somente durante uma janela de tempo específica.
- As **restrições de configuração** incluem requisitos do ambiente de processo. Por exemplo, dados que devem estar disponíveis para um processo, incluindo arquivos, programas, perfis de usuário e dados de conta, como nomes de usuário e senhas.

## Fase II: Design e implementação

Na fase de criação do desenvolvimento de processo, mapeie os requisitos para os objetos de automação e a funcionalidade de operador do CA Process Automation. As etapas do design e da implementação de um processo incluem:

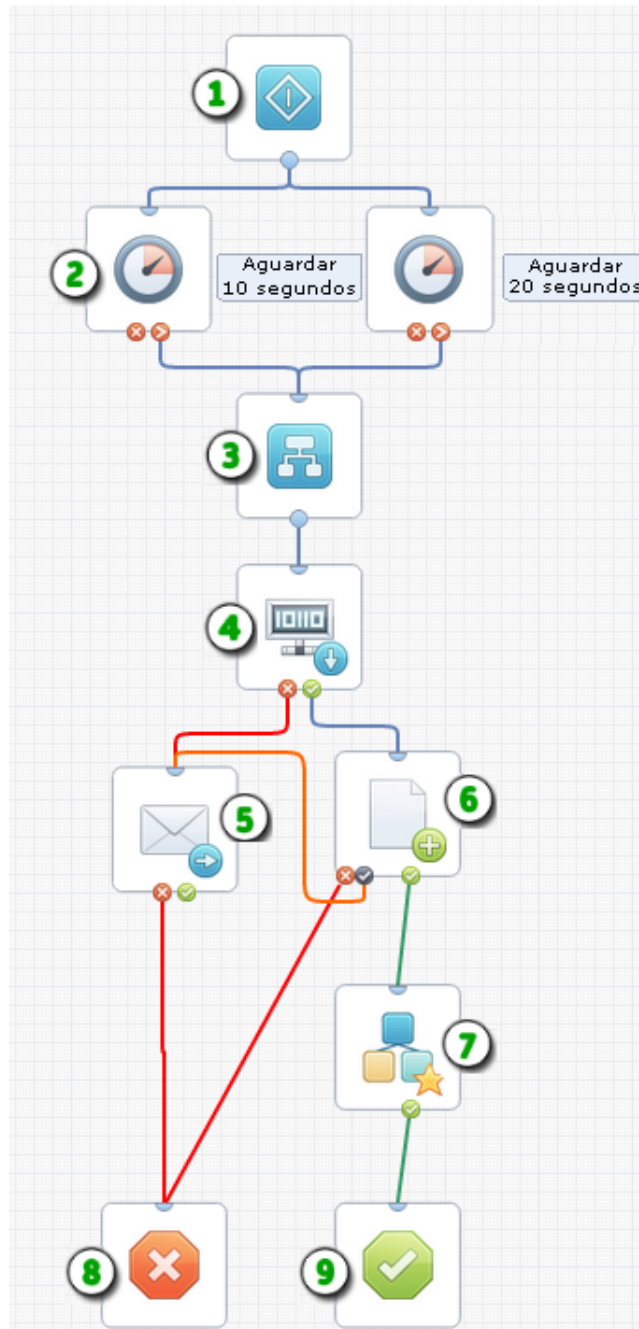
- Identificar os objetos de automação e os operadores de processo para implementar o processo.
- Configurar todas as etapas lógicas e de operadores, incluindo todos os parâmetros, tarefas, resultados, recursos e dependências. Definir as informações necessárias de configuração em tempo de execução (como nomes de usuário, senhas e nomes de arquivo).
- Otimizar o design para reutilização da modularidade e de componentes.
- Definir o início e o monitoramento do processo, incluindo as permissões para iniciar, monitorar e modificar o processo.
- Definir os alertas e determinar quais usuários ou grupos serão alertados.

## Identificar operadores e objetos de processos


Cada processo ou subprocesso que você identifica corresponde a um objeto de processo no Navegador da biblioteca. Os objetos de processo definem uma sequência de etapas vinculadas. Um operador de processo representa cada etapa. Após mapear um processo passo a passo, identifique os operadores que executam cada etapa. Às vezes, um único operador pode executar uma determinada etapa. Outras vezes, uma etapa pode exigir dois ou mais operadores separados.

Defina as etapas em um processo posicionando os operadores com base em ícone que representem ações que o CA Process Automation executa. Os operadores Iniciar começam os processos. Os operadores Interromper encerram os processos. Os operadores lógicos e de controle definem os pontos de início e encerramento, a ramificação e as iterações em um processo. A criação de um processo revela sua estrutura e também mapeia, sincroniza e define a sequência e as dependências entre as tarefas.

### **Exemplo: operadores em um processo**



Item:	Descrição:
1	<b>Iniciar:</b> o operador Iniciar representa o único ponto de entrada em que o processo é iniciado. Os operadores Iniciar também indicam a primeira etapa em uma ramificação independente.

Item:	Descrição:
2	<b>Atrasar:</b> o operador Atrasar atrasa o processamento de ramificações subsequentes de um processo. até que um intervalo especificado de dias, horas, minutos e segundos tenha se passado.
3	<b>Operador Ou:</b> os operadores <i>Ou</i> e <i>E</i> controlam as etapas logicamente. O operador <i>Ou</i> só é concluído (e permite que o processamento continue) quando um dos operadores de entrada é concluído. O operador <i>E</i> só é concluído (e permite que o processamento continue) quando todos os operadores de entrada são concluídos.
4	<b>Obter variável de SNMP:</b> esse operador retorna o valor de uma variável de SNMP.
5	<b>Enviar email:</b> use esse operador para notificar outros usuários por email.
6	<b>Gravar arquivo:</b> esse operador grava uma variável do conjunto de dados em um arquivo. A porta de saída personalizada  especifica um curso de ação predefinido para uma condição de saída específica.
7	<b>Iniciar processo:</b> esse operador inicia outro processo. Você pode definir uma propriedade Loop opcional para executar outro processo repetidamente.
8	<b>Interrupção com falha:</b> o operador Interrupção com falha interrompe todas as ramificações de um processo e define o estado do processo como <i>Com falha</i> . Use um operador Interrupção com falha para o resultado anormal de um processo ou uma ramificação.
9	<b>Interrupção bem-sucedida:</b> o operador Interrupção bem-sucedida interrompe todas as ramificações de um processo e define o estado do processo como <i>Concluído</i> . Use um operador Interrupção bem-sucedida para o resultado normal de um processo ou uma ramificação.

## Configurar os operadores e as etapas

Após identificar os operadores de cada etapa, defina suas propriedades. Pense nos tipos de dados que eles precisam e na origem desses dados. Algumas propriedades do operador são comuns a todos os operadores e outras são limitadas a tipos específicos de operadores. Por exemplo, um operador de banco de dados exige parâmetros para o tipo de banco de dados, o nome do banco de dados, um nome de usuário válido e uma senha.

Você pode definir o valor de uma propriedade para uma entidade literal como *5* ou *Austrália*, ou para uma expressão como *Var1.quantity* ou *GetCountry*. Muitas vezes, as expressões incluem variáveis ou funções. Uma variável armazena um valor. Por exemplo, uma variável pode conter uma senha ou uma configuração de limite. Uma função verifica qualquer instrução lógica e retorna um valor. Por exemplo, uma função pode obter ou definir um valor de campo.

As expressões podem incluir qualquer um dos itens a seguir:

- Variáveis de sistema e do usuário
- Funções
- Valores literais
- Operadores lógicos, de sequência de caracteres, aritméticos e de comparação
- Expressões JavaScript

Alguns campos aceitam somente valores literais. Os rótulos para esses campos são exibidos em itálico. A maioria dos campos oferece suporte a expressões calculadas.

**Observação:** as versões anteriores do CA Process Automation identificam os campos que aceitam expressões calculadas com um asterisco (\*).

Os operadores também compartilham as seguintes propriedades comuns:

- Destino da execução: *onde esse operador deve ser executado?*
- Configurações do tempo limite: *o que deve acontecer quando o operador demora muito tempo para ser executado?*

Para saber mais sobre a funcionalidade e as propriedades de cada operador, consulte a *Referência do criador de conteúdo*.

Os conjuntos de dados armazenam variáveis do CA Process Automation. É possível adicionar e editar variáveis em qualquer um dos três tipos de conjunto de dados:

- **Conjuntos de dados nomeados:** os objetos do conjunto de dados que você cria e mantém no Navegador da biblioteca.
- **Conjuntos de dados do processo:** o conjunto de dados para cada processo.
- **Conjuntos de dados do operador:** o conjunto de dados de cada operador em um processo.

Além disso, o conjunto de dados somente leitura do sistema disponibiliza dados do sistema para qualquer expressão. Use variáveis e expressões mais complicadas para definir os valores que podem ser alterados, como credenciais de usuário. Você pode atribuir valores para expressões e variáveis globalmente ou alterar dinamicamente um valor no código. Use valores literais somente para configurar parâmetros se os valores não forem alterados.

## Conjuntos de dados nomeados

Use variáveis globais em conjuntos de dados nomeados a fim de definir as configurações de parâmetro. Os conjuntos de dados nomeados são objetos de automação que definem variáveis com escopo na biblioteca inteira. Dependendo das configurações de segurança para um objeto de conjunto de dados, as variáveis estarão globalmente acessíveis a qualquer expressão em qualquer objeto de automação. Um campo em outro conjunto de dados pode fazer referência a uma variável do conjunto de dados.

Use conjuntos de dados nomeados para especificar informações, como contas e senhas usadas em vários processos. Quando as informações salvas em um conjunto de dados nomeado são alteradas, é necessário alterá-las apenas uma vez no conjunto de dados nomeado, e não em cada objeto de automação que as utiliza.

Você pode usar Formulários de solicitação inicial para coletar informações de usuários autorizados e, em seguida, atualizar os valores das variáveis do conjunto de dados. É possível usar um operador Executar o JavaScript em um processo para calcular e armazenar valores nas variáveis nomeadas.

## Recursos

Alguns operadores ou processos utilizam muita memória, tempo de processador ou outros recursos limitados. Por exemplo, se um processo filho solicitar outros processos, monopolizando muitos ciclos de CPU, você poderá limitar o número de instâncias do processo filho em execução em qualquer momento específico.

Para gerenciar recursos, defina um objeto de recurso utilizado pelo processo. Sempre que o recurso estiver bloqueado ou sendo consumido por processos em execução, solicitações adicionais para o recurso deverão aguardar até que o recurso esteja livre ou disponível. Use recursos para lidar com o balanceamento de carga em seus processos.

### Mais informações:

[Recursos](#) (na página 369)

## Atribuir condições de saída para os resultados

Você pode definir uma condição de saída para cada resultado do operador. Um operador pode ter várias condições de saída para resultados diferentes, como *Concluído* ou *Falha*.

Operadores oferecem suporte a condições de saída específicas. Alguns operadores também permitem a definição de uma porta personalizada. Por exemplo, você pode definir uma porta personalizada que será executada se o resultado do operador for *Verdadeiro*, *Falso*, "azul", 500, -2 ou algum outro valor designado.

Defina uma porta personalizada quando você tiver um curso de ação distinto para uma determinada condição de saída. Uma condição assim é quando há falha na importação de um banco de dados por algum motivo específico. Quando um processo é executado e não há condição de saída específica, o processo é suspenso e alternado para um estado bloqueado, ação de usuário pendente. Um administrador ou outro usuário com permissões suficientes pode alterar as configurações de parâmetro, se necessário, e reiniciar um processo suspenso do ponto em que ele falhou ou desde o início.

## Otimizar para reutilização da modularidade e de componentes

Divida os processos em subprocessos separados ou objetos de processo *filho*. Use um processo *pai* principal para controlar os subprocessos. Cada subprocesso executa uma tarefa distinta ou um conjunto de tarefas, como extrair dados de um banco de dados e gravá-los em arquivos separados. Quando vários processos compartilharem um subprocesso comum, mantenha a flexibilidade no design para permitir que o subprocesso continue a ser usado nos diferentes processos pai.

Adicione flexibilidade usando variáveis ou expressões para definir parâmetros no processo e em cada operador. Dessa forma, se um processo for reutilizado em diferentes ambientes ou sistemas, ajustar o processo é tão simples quanto atualizar apenas as variáveis relevantes.

O CA Process Automation também ajuda a controlar o fluxo entre diferentes caminhos ou subprocessos. Por exemplo, quando os processos exigem entrada no tempo de execução, você pode usar os *formulários de solicitação de interação* para solicitar informações aos usuários. Use um ou mais formulários de solicitação de interação para implementar pontos de verificação de processo onde usuários ou grupos específicos são necessários para autorizar um subprocesso antes de continuar. O operador Atribuir tarefa de usuário em um objeto de processo pausa o processo e, em seguida, abre o formulário. O usuário poderá então digitar os valores necessários.

## Operadores personalizados

É possível basear um objeto de operador personalizado em qualquer um dos outros operadores disponíveis em seu ambiente. Use os operadores personalizados para:

- **Facilitar a reutilização:** use o mesmo operador com as mesmas definições de configuração em diferentes processos.
- **Salvar as configurações para uma tarefa:** você pode pré-configurar operadores personalizados para executar tarefas específicas. Por exemplo, é possível definir um operador personalizado para trabalhar com um aplicativo que já está instalado no sistema.
- **Interagir com aplicativos empresariais:** você pode exportar uma pasta como pacote de conteúdo para oferecer suporte a aplicativos empresariais comuns.

É possível modificar seu operador personalizado sem alterar o operador original.

## Definir o início e o monitoramento do processo

### Início do processo

Ao criar processos, decida como cada processo será iniciado. Você pode criar um processo para iniciar de várias maneiras:

- **Início manual:** você pode iniciar manualmente um processo no Navegador da biblioteca ou no Criador de processos. Como um criador e desenvolvedor de conteúdo trabalhando no CA Process Automation, você inicia processos repetida e manualmente, ao mesmo tempo que os cria e os testa.
- **Início automático:** é possível construir um processo que inicie automaticamente outro processo usando o operador Iniciar processo.
- **Início automático por entidade externa:** você pode definir um processo para ser iniciado de um sistema ou aplicativo externo usando serviços web, um utilitário de linha de comando, ou *disparadores*. Por exemplo, é possível especificar que outro programa inicie um processo usando serviços web (*executeProcess* ou *executeStartRequest*). Você pode definir um processo para ser iniciado por um evento externo por meio do uso de disparadores com suporte, incluindo criação de arquivo, email de entrada, SNMP traps e eventos de conectores do UCF.
- **Iniciar por programação:** você pode programar um processo para ser iniciado usando um objeto Programação.
- **Iniciar por usuário e formulário:** você pode criar um objeto Formulário de solicitação inicial ou objeto Formulário de solicitação de interação que pode solicitar que um usuário responda ao formulário resultando na iniciação de outro processo. Um exemplo desse método para iniciar um processo é um formulário de envio de relatório de despesas. Um funcionário pode preencher o formulário e iniciar um processo *UpdateExpenses* sob demanda.

## Controle de processo

Criadores, administradores e a equipe de produção usam o CA Process Automation para controlar processos. Para obter mais informações, consulte o seguinte conteúdo para a sua função:

- **Criadores:** para executar, testar e depurar processos, consulte [Executando, testando e depurando processos](#) (na página 419).
- **Administradores:** para administrar o conteúdo do sistema ou aplicativo, consulte o *Guia do Administrador de Conteúdo do CA Process Automation*.
- **Usuários de produção:** para iniciar processos de maneira interativa em um ambiente de produção, consulte o *Guia do Usuário de Produção do CA Process Automation*.

## Processos de programação

Os dois objetos de automação usados para a programação de processos são *calendários* e *programações*. Use objetos de calendário para definir condições de data para iniciar tarefas que lançam processos ou operadores individuais. Use objetos de *programação* para programar tarefas usando os objetos de calendário e/ou datas explícitas.

- A criação de condições de data com objetos de calendário é descrita em [Calendários](#) (na página 381).
- A programação de tarefas com objetos de programação é descrita em [Programações](#) (na página 397).

**Mais informações:**

[Calendários](#) (na página 381)

## Processos sob demanda

As ferramentas administrativas e de desenvolvimento disponíveis na guia Operações expõem a estrutura subjacente e a lógica de bibliotecas e aplicativos. A guia Operações pode exibir um objeto Formulário de solicitação inicial que solicita informações a um usuário necessárias para iniciar um processo. Os usuários autorizados podem usar Formulários de solicitação inicial para executar processos sob demanda sem precisar conhecer os detalhes técnicos relacionados a:

- Como o CA Process Automation funciona.
- Como a arquitetura do sistema foi criada.
- Como cada processo está estruturado.

**Mais informações:**

[Formulários de solicitação inicial](#) (na página 275)

## Processos acionados

O CA Process Automation oferece suporte a eventos e *disparadores* que podem iniciar processos a partir de sistemas e aplicativos externos. Você pode designar aplicativos, mensagens de email, páginas da web ou outros processos para disparar um processo. Qualquer um dos métodos a seguir pode ser usado para disparar um processo:

- FTP
- Registro em HTTP/SOAP
- SNMP traps personalizadas
- SMTP (email)
- Evento do UCF

Para obter mais informações, consulte [Disparadores](#) (na página 410).

## Monitorar processos

Você pode monitorar graficamente os processos usando um objeto *Exibição de processos*. Os objetos Exibição de processos fornecem portais ou atalhos a todos os processos e objetos relacionados de uma determinada categoria de propriedade. Por exemplo, uma equipe de repositório de dados necessita de acesso a um objeto Exibição de processos que contém atalhos para todos os processos de extração, transformação e carregamento (ETL) a fim de preencher os repositórios de dados.

Considere as funções e a propriedade ao gerenciar uma exibição de processos. Por exemplo, um usuário final normalmente deve atender e responder aos processos à medida que eles ocorrerem. Por outro lado, um administrador pode precisar verificar o histórico. Uma exibição de processos leva em conta a propriedade, incluindo apenas os objetos que a pessoa que os está usando deve monitorar.

**Mais informações:**

[A página Operações](#) (na página 421)

## Definir permissões

As permissões dependem de quem possui os componentes que estão sendo gerenciados ou quem é responsável por determinadas funções de gerenciamento. É possível atribuir permissões por grupo funcional (como administradores, operadores ou uma equipe de repositório de dados) ou por propriedade. Em seguida, você pode adicionar ou remover usuários individuais de um grupo.

Por exemplo, uma equipe de repositório de dados pode monitorar os processos de extração, transformação e carregamento (ETL) para o preenchimento de um repositório de dados. Crie um grupo *DW* contendo os integrantes da equipe de repositório de dados. Em seguida, atribua as permissões Listar e Abrir a esse grupo para um objeto Exibição de processos, fornecendo todos os atalhos relacionados aos processos de ETL.

Aos administradores são atribuídas permissões maiores que os usuários gerais. Um administrador de banco de dados pode ter controle sobre os processos para atualizar um repositório de dados ou restaurar e fazer backup de determinados bancos de dados. Se você estiver fazendo o desenvolvimento para um ambiente empresarial complicado com várias funções e propriedade diferentes, considere essa abordagem. Mapeie a propriedade em uma planilha ou em outro formato para os grupos, usuários e funções, de modo a atribuir as permissões apropriadas.

A propriedade para um processo (ou qualquer objeto de automação) é inicialmente atribuída ao usuário para o qual ela foi criada. A propriedade pode ser alterada.

A definição de permissões para as ações dos usuários em tempo de execução é feita pelo editor de permissões no CA EEM. Os integrantes do grupo *PAMAdmins* padrão têm permissões completas em todas as pastas e em todos os objetos de automação. Os integrantes do grupo *PAMUsers* padrão não têm permissões em nenhum objeto de automação ou pasta. Os criadores de conteúdo normalmente são integrantes do grupo Criadores.

Um administrador de conteúdo ou o proprietário de uma pasta pode atribuir permissões para você na pasta com a propriedade Definir proprietário. As permissões de pasta são herdadas pelos objetos de automação adicionados à pasta. Um administrador com credenciais do CA EEM pode atribuir permissões para você em pastas e objetos de automação no CA EEM.

**Observação:** consulte o *Guia do Administrador de Conteúdo* para obter detalhes.

## Definir alertas

É possível usar alertas de email em processos para notificar erros ou incidentes a usuários ou administradores, ou para informar os usuários que um processo ou uma tarefa foi concluída.

Ao criar alertas em um processo, defina primeiro os principais destinatários. Por exemplo, você poderá notificar um único administrador quando um processo for bem-sucedido, falhar ou estiver aguardando a entrada do usuário. Também é possível criar um grupo de destinatários para alertar os gerentes que talvez precisem receber relatórios mensais.

Crie processos separados que não enviem apenas alertas, mas que manipulem o escalonamento e executem outras tarefas relacionadas ao alerta. Em seguida, chame o processo de alerta de outros processos, conforme necessário, usando o operador *Iniciar processo*.

No CA Process Automation, sempre utilize variáveis de conjuntos de dados nomeados para especificar destinatários de email, usuários, grupos, perfis e senhas de conta. Use as variáveis para definir parâmetros no operador *Enviar email*. Quando ocorrerem alterações, você precisará apenas editar as variáveis em um conjunto de dados nomeado, e não em todos os processos que usam os valores.

Como as pessoas mudam, é possível definir grupos de emails de usuários ou aliases de email com base em funções ou cargos. Quando há alterações de pessoal, você poderá alterar pessoas atribuídas ao aliases sem precisar editar as variáveis. Sua organização talvez já tenha aliases para as funções ou grupos apropriados.

## Fase III: Teste e implantação

A implantação do pacote de gerenciamento do CA Process Automation em um ambiente de produção inclui as seguintes atividades:

1. Agrupar os objetos de automação em uma pasta. Os administradores implantam uma pasta em outro orquestrador, em geral, em um ambiente diferente.
2. Testar, depurar e verificar todo o conteúdo para transição.
3. Exportar a pasta como pacote de conteúdo do ambiente de desenvolvimento.
4. Importar o pacote de conteúdo para um ambiente de produção.
5. Fornecer as instruções e treinamento para que os usuários possam ativar programações e usar formulários para iniciar processos sob demanda.

## Testar o processo com objetos relacionados

O teste é essencial para implantar o pacote de gerenciamento do CA Process Automation que representa os processos que você deseja automatizar. Verifique se todos os componentes funcionam em conjunto como previsto antes de implantar o pacote de conteúdo do ambiente de criação para o ambiente de produção.

O CA Process Automation possui vários recursos para testar processos:

- Você pode iniciar um processo no modo *suspense* na paleta do Criador e conduzir testes em tempo de execução. É possível iniciar e pausar a instância do processo à medida que ela é executada.
- É possível definir *pontos de interrupção* para pausar e depurar as etapas do processo.
- Você pode executar processos no modo de *simulação*, que permite definir resultados específicos para uma etapa. Esse modo retorna os resultados simulados de cada etapa sem executar nenhuma ação real.

O teste ocorre nos seguintes níveis:

- Os criadores de conteúdo testam e verificam se cada processo e os objetos de automação utilizados funcionam corretamente no ambiente de criação.
- Os administradores testam a implantação após a exportação e a importação para verificar se cada processo e os objetos de automação utilizados funcionam corretamente no ambiente de produção. Esse processo é transparente quando os administradores criam touchpoints no ambiente de produção com os mesmos nomes que no ambiente de criação. Cada integrante do par de touchpoints define como destinos os agentes em seus respectivos ambientes.

A implantação inteira pode passar em todos os testes e se tornar operacional ao mesmo tempo que um formulário específico ainda está sendo recriado e testado. Os seguintes termos refletem essa diferença:

- *Testes de unidade* em oposição a *testes de integração*
- *Uma build interna de release candidata a lançamento* em oposição a *uma build de prova de conceito do cliente*
- *Desenvolvimento* em oposição à *produção*

Periodicamente, teste novamente os processos críticos à missão que não são executados com frequência (como failovers de sistema). Os testes periódicos verificam se os processos e os usuários permanecem atuais e em funcionamento no ambiente de produção, e impedem que os incidentes se tornem crises.

## Gerar a documentação do processo

Várias funções podem gerar a documentação do processo várias vezes durante o processo de teste e implantação:

- Durante o design, a documentação do processo é útil para que o criador de conteúdos transfira o desenvolvimento de uma automação em andamento para outro criador. O administrador de conteúdo também pode usar a documentação do processo para monitorar o processo de automação.
- No final do processo de criação, os criadores de conteúdo podem gerar a documentação do processo para referência posterior.
- Durante o planejamento de processos subsequentes, os administradores de conteúdo podem consultar a documentação do processo para obter informações sobre conjuntos de dados ou outros componentes reutilizáveis.
- Quando um novo processo for importado para o ambiente de produção, o administrador de produção poderá gerar a documentação do processo para revisão antes de executar o processo pela primeira vez.
- Após usar um processo automatizado por algum tempo, o administrador de produção poderá usar a documentação de produção gerada ao solicitar aprimoramentos. Com o diagrama do fluxo do processo, o administrador pode facilmente indicar, com precisão, onde foi solicitada uma alteração em um processo.

## Definir a versão de release do processo e de objetos relacionados

Você pode definir um valor para o atributo Versão da release na preparação para a implantação de objetos específicos da release em um ambiente de produção. O arquivo de exportação contém apenas a versão selecionada de cada objeto que está sendo exportado. Se você exportar objetos em um pacote de conteúdo, os usuários não poderão modificar suas versões de release após a importação. A versão base de todos os objetos é definida durante a importação. Como resultado, os usuários não poderão modificar a versão importada de nenhum objeto. Se os usuários de produção precisarem modificar um objeto importado, eles deverão salvá-lo como uma nova versão.

## Montar o processo com objetos relacionados

Uma pasta reúne os objetos de automação que estão relacionados a um ou mais processos que estão sendo implantados. Um objeto de pasta inclui objetos utilizados pelo processo (como conjuntos de dados) e objetos que usam o processo (como formulários de solicitação inicial).

## Implantar uma versão da release do processo com objetos relacionados

Depois que os criadores testam um processo no ambiente de criação, eles definem uma versão da release no objeto de processo e em todos os objetos relacionados. Os objetos relacionados incluem todos os objetos que o processo utiliza e todos os objetos que usam o processo.

Um administrador ou criador monta a versão de cada objeto para liberação em uma pasta. Um administrador exporta uma pasta de conteúdo de automação relacionado para um arquivo XML. O administrador então importa esse arquivo XML para o ambiente de produção.

## Especificar instruções de importação

Para ajudar a facilitar a implantação do pacote de conteúdo, forneça instruções de importação. Escreva suas instruções para os administradores do ambiente de produção e para qualquer outro usuário do processo. Indique como os elementos do processo devem ser configurados para funcionar no ambiente de produção.

Essas instruções podem incluir os seguintes tópicos de exemplo:

- Como instalar atualizações ou aplicativos necessários
- Como configurar nomes de usuário, grupos e senhas
- Como definir permissões para usuários ou grupos
- Como configurar conjuntos de dados para o ambiente de produção
- Como configurar o hardware adicional para operadores
- Como ativar programações
- Como configurar os disparadores externos
- Como usar um formulário para iniciar processos sob demanda



# Capítulo 2: Guia de Introdução

---

Esta seção o orienta pelo CA Process Automation e fornece informações básicas sobre como efetuar logon e familiarizar-se com a interface de usuário.

Esta seção contém os seguintes tópicos:

[Vá até o CA Process Automation e efetue logon.](#) (na página 34)

[Definir as configurações do usuário](#) (na página 35)

[A interface do usuário do CA Process Automation](#) (na página 36)

[Explorar o conteúdo pronto para uso](#) (na página 55)

[Usar o CA EEM para alterar sua senha do CA Process Automation](#) (na página 56)

[Navegadores da web](#) (na página 57)

## Vá até o CA Process Automation e efetue logon.

O URL usado para acessar o CA Process Automation depende se o orquestrador de domínio está configurado com um nó (não agrupado) ou vários nós (agrupado). É possível navegar diretamente para um CA Process Automation não agrupado. Para um CA Process Automation agrupado, procure o balanceador de carga associado. É possível acessar todos os orquestradores no domínio iniciando o URL para o orquestrador de domínio ou para o balanceador de carga para o orquestrador de domínio.

### Siga estas etapas:

1. Procure o CA Process Automation.
  - Para uma comunicação segura, use a sintaxe a seguir:  
`https://server:port/itpam`

#### Exemplos:

`https://Orchestrator_host:8443/itpam`  
`https://loadBalancer_host:443/itpam`

- Para uma comunicação básica, use a sintaxe a seguir:  
`http://server:port/itpam`

#### Exemplos:

`http://Orchestrator_host:8080/itpam`  
`http://loadBalancer_host:80/itpam`

A página de logon do CA Process Automation é exibida.

2. Digite as credenciais de sua conta de usuário.

**Observação:** se o CA EEM estiver configurado para fazer referência a usuários de vários Microsoft Active Directories e o CA Process Automation não aceitar seu nome de usuário não qualificado, digite o nome de sua entidade principal, que é *nome\_do\_domínio\nome\_do\_usuario*.

3. Clique em Efetuar logon.

O CA Process Automation é exibido. A guia Início é exibida.

## Definir as configurações do usuário

Após efetuar login pela primeira vez, defina as configurações de exibição da interface do usuário.



### Siga estas etapas:

1. Na lista de links, na parte superior da página principal, clique em seu nome de usuário.
2. Na caixa de diálogo Configurações do usuário, indique suas preferências de exibição para o formato de data e hora, e idioma.
3. Clique em Salvar e fechar.

**Observação:** o CA Process Automation salva e restaura automaticamente outros ajustes personalizados feitos por você durante o seu trabalho. Por exemplo, se você encaixar uma paleta ou personalizar as colunas que deseja exibir em uma tabela ou lista.

## A interface do usuário do CA Process Automation

A interface do CA Process Automation fornece um ambiente administrativo e de desenvolvimento integrado para exibir, gerenciar e executar todos os objetos em seus sistemas de automação. O CA Process Automation é um aplicativo web que pode ser aberto em qualquer computador que tenha acesso ao orquestrador do CA Process Automation.

Cada guia principal na parte superior da página apresenta uma seção ou uma área funcional exclusiva do aplicativo. Controles comuns em todo o aplicativo facilitam a sua utilização. Por exemplo, você usa as mesmas etapas básicas para classificar uma lista de entradas e configurar quais colunas devem ser exibidas.

*Equation 1: Este gráfico destaca como trabalhar com a interface de usuário para executar as funções do CA Process Automation.*

The screenshot displays the CA Process Automation interface. On the left is a tree view of organizational objects, with 'Live Objects' selected under 'Automation Library' in 'Project P01'. The main area shows a table of objects with columns for 'Nome', 'Tipo', and 'Proprietário'. A dialog box titled 'Definir proprietário' is open, showing a list of available users with 'Henry Hoevak' selected.

Nome	Tipo	Proprietário
CCalendar_001	Calendário	pamadmin
CCalendar_002	Calendário	User Z0450
CCalendar_003	Calendário	User Z0250
CCustomIcon_001	Ícone personalizado	User X0300
CCustomIcon_002	Ícone personalizado	User Z0300
CCustomIcon_003	Ícone personalizado	User Z0250
CCustom_Operator_001		
CCustom_Operator_002		
CCustom_Operator_003		
CCustom_Operator_004		
CDataset_001		
CDataset_002		
CDataset_003		
CDataset_004		
CIRF_001		
CIRF_002		
CIRF_003		
CPack_001		
CProcess_Watch_001	Exibição de processos	pamadmin
CProcess_Watch_002	Exibição de processos	pamadmin
CResources_001	Recursos	pamadmin
CResources_002	Recursos	pamadmin
CSRF_001	Formulário de solicitação inicial	User Z0350

ID do usuário	Nome do usuário
hhoevak	Henry Hoevak
aprasad	Amanda Prasad
rjones	Ron Jones

Item:	Descrição:
1	<b>Controles do sistema operacional e do navegador:</b> embora não faça parte do CA Process Automation, o seu sistema operacional fornece controles para trabalhar com a janela atual para tarefas como Minimizar; Maximizar, Restaurar e Fechar. O navegador também exibe seus próprios menus, barras de ferramentas, painéis e áreas de pesquisa. Às vezes é possível usar recursos do navegador para complementar recursos do CA Process Automation incorporados, como a atualização de uma página ou o ajuste da ampliação da exibição (Zoom).
2	<b>Links:</b> o CA Process Automation fornece links de aplicativo comuns, incluindo Configurações de usuário, Ajuda e Logoff. As páginas individuais incluem links adequados ao conteúdo relacionado.
3	<b>Principais guias do aplicativo:</b> clique em uma guia para focalizar uma seção específica do CA Process Automation. A guia Biblioteca é selecionada na imagem de exemplo, para que o aplicativo exiba pastas e objetos no Navegador da biblioteca.
4	<b>Barra de ferramentas:</b> muitas páginas e caixas de diálogo exibem barras de ferramentas específicas com botões e ícones de ferramentas apropriados.
5	<b>Painéis:</b> os <i>painéis</i> dividem uma janela ou página. Neste exemplo, o painel Navegador da biblioteca é exibido à esquerda e apresenta um campo de entrada de filtro e uma hierarquia de pastas expansível. A página principal exibe informações detalhadas sobre a entrada selecionada em um painel.
6	<b>Página principal:</b> a área principal de uma página exibe informações essenciais sobre um item selecionado em um painel. O CA Process Automation geralmente apresenta os dados em tabela, lista, formulário, tela de criação ou gráfico. Quando necessário, a área principal é ainda dividida em paletas, guias, painéis ou outros controles visuais.
7	<b>Caixa de diálogo:</b> quando você clica em determinados botões, aplica ações ou emite comandos, o aplicativo muitas vezes apresenta caixas de diálogo para coletar entradas adicionais.
8	<b>Mensagem:</b> a lógica predefinida e os eventos que você ou o sistema inicia pode resultar no aparecimento de mensagens. A maioria desses alertas é informativa e fornece o feedback necessário. No entanto, algumas mensagens exibem avisos críticos criados para proteger seus dados. As mensagens de erro fornecem informações úteis que podem ser combinadas com dados do arquivo de log para solucionar problemas.


## Páginas principais do aplicativo


Clique em qualquer uma das seis guias principais para navegar nas páginas principais do aplicativo. À medida que você trabalha, o CA Process Automation aplica lógica e permissões para determinar o que poderá ser exibido. Às vezes, o CA Process Automation altera automaticamente a seleção de guia atual ou abre outra janela. Por exemplo, ao abrir um processo no Navegador da biblioteca, você vê o processo automaticamente no Criador. Quando você abre um objeto de programação a partir da guia Biblioteca, é exibida a janela Editor de programação. Ao clicar em uma instância de processo na guia Operações, uma janela Instâncias de processo separada é exibida automaticamente.

## Página inicial

Use essa página como um ponto de partida conveniente ou um painel pessoal para sua sessão de trabalho do CA Process Automation. Os recursos disponíveis são realçados no exemplo a seguir.


Firefox ▾


CA Process Automation
1




**Criar um processo**

Clique aqui para iniciar a criação de um processo usando o criador arrastar e soltar.




**Acessar o Cloud Commons**

Acesse o Cloud Commons, o marketplace online do conteúdo do CA Process Automation.



**Explorar o conteúdo pronto**

Conteúdo que demonstra os casos de uso do CA Process Automation.



**Visualizar tutoriais**

Permite visualizar os tutoriais para implementar casos de uso do CA Process Automation.

2

**Minhas tarefas (Atualizar)**

	Status	Título	Descrição	Data de vencimento
✓	Concluído	TaskName1	Lorem ipsum dolor sit amet	
⊖	Pendente	TaskName2	Lorem ipsum dolor sit amet	
⊖	Pendente	TaskName3	Lorem ipsum dolor sit amet	

3

**Itens reservados por mim (Atualizar)**

Data	Nome	Descrição
23/04/2012 17:19:25	Process Watch	
23/04/2012 17:19:25	Process	
23/04/2012 17:19:25	Lokesh	
23/04/2012 17:19:25	Interaction Request Form_4	
23/04/2012 17:19:25	Chisu	

**Item: Descrição:**

- 1 **Guia Início:** clique na guia Início para exibir a página inicial. A página inicial é exibida quando você efetua logon no CA Process Automation.
- 2 **Botões rápidos:** o CA Process Automation fornece opções para ajudar a economizar tempo e começar a trabalhar. O canto inferior direito da página também lista links rápidos.
- 3 **Tarefas e objetos de automação:** o CA Process Automation exibe tabelas que listam suas tarefas e os objetos de automação que você reservou (isto é, as versões de trabalho dos objetos).
- 4 **Minha atividade recente:** uma lista de atividades de processos recentes é exibida classificada por data e hora. Clique em um link para abrir a instância do processo.

**Navegador da biblioteca**

Para gerenciar objetos de automação em pastas, use o Navegador da biblioteca.

The screenshot displays the CA Process Automation interface. The top navigation bar includes tabs for 'Início', 'Biblioteca', 'Criador', 'Operações', 'Configuração', and 'Relatórios'. The 'Biblioteca' tab is active. On the left, the 'Orchestrator' pane shows a tree view of folders like 'Draft Objects', 'My Objects', and 'Organizational Objects'. A context menu is open over the 'Automation Library' folder, showing options like 'Novo objeto', 'Copiar', and 'Excluir'. The main area shows a table of objects with columns: Nome, Tipo, Proprietário, Caminho, Data de modificação, Reservado, Versão atual, and Versão máxima. A table with 4 columns is visible below the main table, showing update history.

Última atualização	Nome de usuário	Tipo de ação	Versão
23/04/2012 16:40:17	Designer_72	Palavras-chave atualizadas.	2
23/04/2012 16:40:07	username	Palavras-chave atualizadas.	2
18/04/2012 14:38:55	username	Programação ativada.	2
18/04/2012 14:38:38	ehoon	Disponibilizado com a nova versão.	2
18/04/2012 14:38:38	ehoon	Conjunto de versões atuais	2

**Item: Descrição:**

- 1 **Guia Biblioteca:** clique nessa guia para ir para o Navegador da biblioteca. Use o Navegador da biblioteca para visualizar, criar, editar ou remover objetos de automação e pastas.

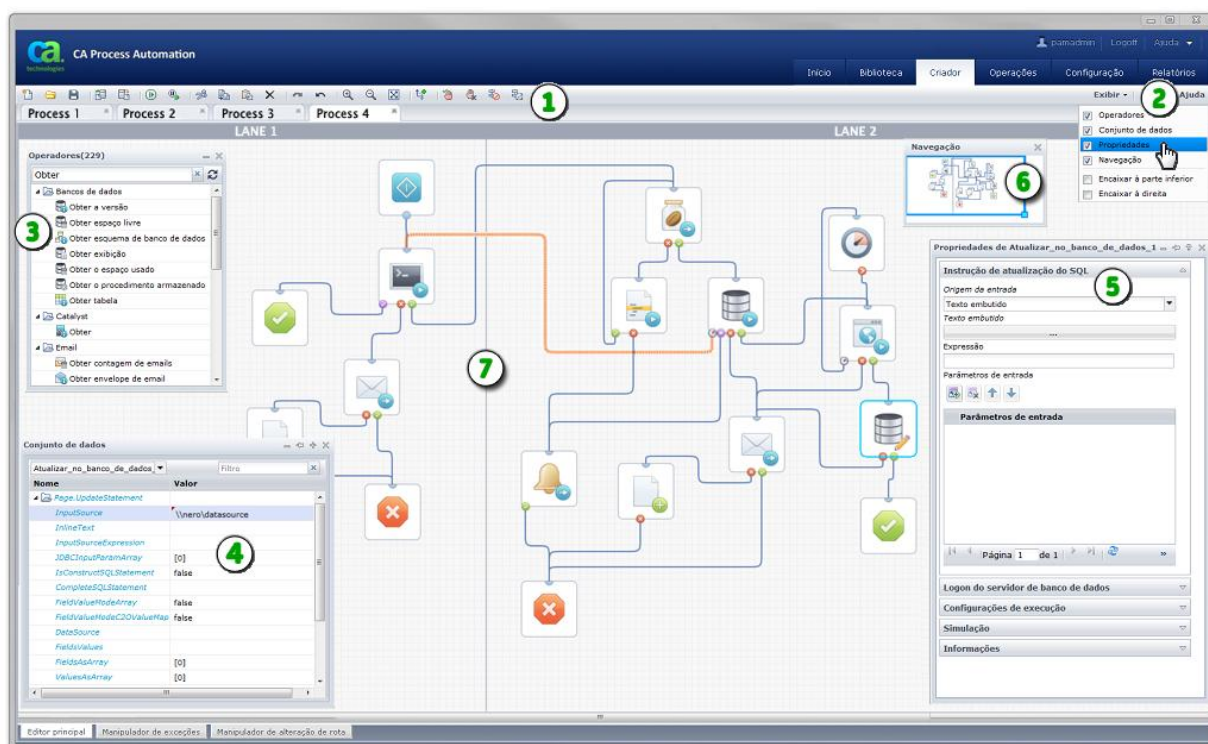
Item:	Descrição:
2	<b>Painel de pastas:</b> o CA Process Automation exibe os orquestradores e as pastas que você pode acessar nesse painel redimensionável. Aplique um filtro para exibir somente pastas correspondentes. Por exemplo, <i>Folder_2</i> localizaria Folder_2, Folder_200 e MyFolder_2.
3	<b>Área de pesquisa:</b> insira os critérios para executar uma pesquisa básica ou avançada. Por exemplo, localize um ou mais objetos por nome, tipo ou palavra-chave. Também é possível pesquisar e substituir objetos.
4	<b>Criar objetos de automação:</b> no Navegador da biblioteca, é possível criar pastas para armazenar objetos de automação. Clique com o botão direito do mouse em uma pasta para ver um menu de comandos que podem ser executados nessa pasta. Também é possível criar novos objetos de automação, como processos, programações ou um formulário de solicitação inicial.
5	<b>Renomear um objeto:</b> clique com o botão direito do mouse em um objeto e selecione Renomear no menu de atalho para editar seu nome.
6	<b>Editar um objeto:</b> use a barra de ferramentas ou clique com o botão direito do mouse em um objeto para exibir um menu de comandos.
7	<b>Propriedades:</b> esse painel com guias exibe dados gerais, tags, versões, release e uma trilha de auditoria de cada objeto.
8	<b>Lixeira:</b> use a lixeira para gerenciar objetos e pastas que foram excluídos. O aplicativo permite restaurá-los, deixá-los na lixeira ou limpá-los permanentemente.

**Observação:** para obter informações sobre como configurar a segurança de objetos e pastas, consulte o *Guia do Administrador de Conteúdo*.

## Designer

Para criar, editar e testar um processo, use a página Criador. O Criador oferece suporte aos seguintes detalhes do processo:

- Fluxo básico do operador
- Documentação de objetos do processo
- Resultados lógicos
- Linhas de conexão
- Navegação em conjuntos de dados
- Definição de propriedades
- Criação de código
- Monitoramento de instâncias do processo
- Percorrer, depurar, testar e controlar instâncias do processo



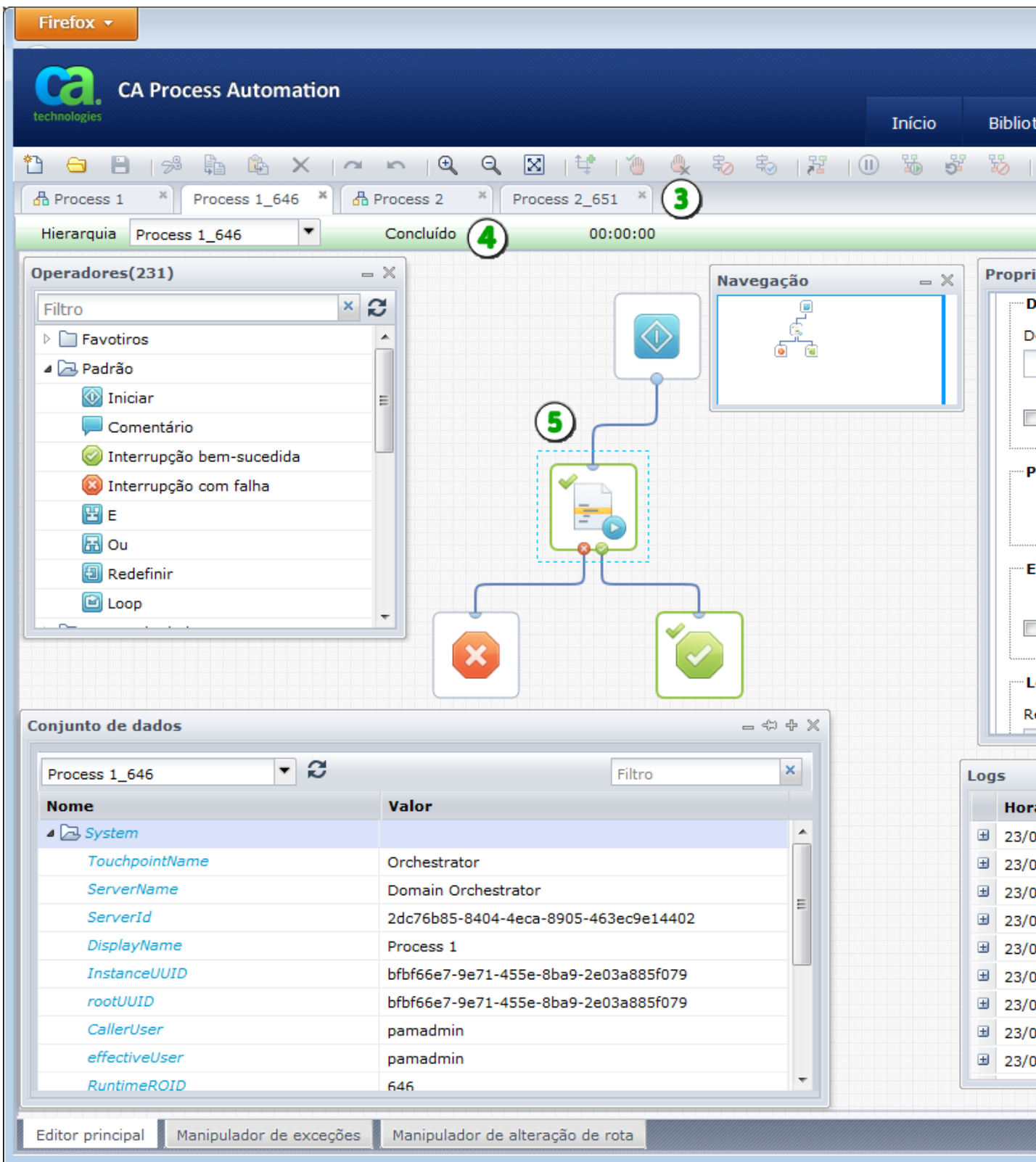
### Item: Descrição:

1

**Guia Criador:** quando você abrir um novo processo ou editar um processo existente do Navegador da biblioteca, essa guia será exibida. Se essa guia não estiver disponível, significa que você ainda não abriu um processo ou não tem permissões de segurança para abrir um processo.

Item:	Descrição:
2	<b>Barra de ferramentas Criador de processos:</b> use os botões de ferramentas para criar e testar processos. Use as configurações de Exibir à direita para mostrar ou ocultar as paletas Operadores, Conjunto de dados, Propriedades e Navegação.
3	<b>Guias do processo:</b> cada processo aberto aparece em sua própria guia. Você pode copiar e colar objetos entre as guias.
4	<b>Criador de processos:</b> o design do processo real é exibido nessa área de trabalho, tela ou layout. O Criador de processos inclui a grade, as rotas, todos os operadores, portas e linhas de conexão.
5	<b>Paleta Operadores:</b> arraste e solte os operadores com funções específicas a partir dessa paleta até o layout do seu processo.
6	<b>Paleta Conjunto de dados:</b> use essa paleta para exibir ou editar as variáveis no processo ou nos conjuntos de dados do operador.
7	<b>Paleta Propriedades:</b> use esta paleta e seus links e janelas para gerenciar as propriedades de um operador. Este exemplo exibe propriedades do operador Executar o script. O usuário também adicionou um código na caixa de diálogo Código de execução posterior.
8	<b>Paleta Navegação:</b> use esta paleta para navegar para regiões específicas em processos maiores. Para economizar tempo, tente efetuar um movimento panorâmico nessa paleta em vez de rolar o layout do criador principal.

Você também pode executar as instâncias de processo na página Criador.



Item:	Descrição:
1	<b>Guia Criador:</b> esta guia é exibida quando você abre uma instância existente do processo a partir de outra página no aplicativo. Se essa guia não estiver disponível, significa que você ainda não abriu uma instância de processo ou não tem permissões de segurança para abrir uma instância.
2	<b>Barra de ferramentas da instância de processo:</b> use os botões de ferramentas para interromper, iniciar, depurar e testar as instâncias reais de um processo. Use as configurações de Exibir à direita para mostrar ou ocultar as paletas Operadores, Conjunto de dados, Propriedades, Navegação e Logs.
3	<b>Guias Processo e Instância:</b> cada processo e cada instância de um processo que você abrir serão exibidas em sua própria guia. As guias de instância de processo são exibidas próximas à guia do criador do processo de origem. As guias do processo exibem um ícone. As guias de instância de processo não exibem nenhum ícone. O CA Process Automation atribui um número de ID de processo exclusivo a cada nome de instância para ajudar a identificar as instâncias diferentes. Você pode copiar e colar objetos entre as guias.
4	<b>Barra de status da instância de processo:</b> essa barra exibe o status da instância. Quando aplicável, a duração de status também é exibida. Por exemplo, uma instância Aguardando também mostrará um relógio ao vivo indicando por quanto tempo a instância permaneceu nesse estado. Você também pode usar o controle Hierarquia para focalizar subprocessos específicos.
5	<b>Instância de processo:</b> a criação da instância de processo é exibida nessa área de trabalho, tela ou layout. A instância de processo inclui a grade, as rotas e todos os operadores, portas e linhas de conexão. Use-a para rastrear o caminho do processo durante sua execução.
6	<b>Paleta Logs:</b> use essa paleta para verificar ou solucionar problemas das instâncias de processo.

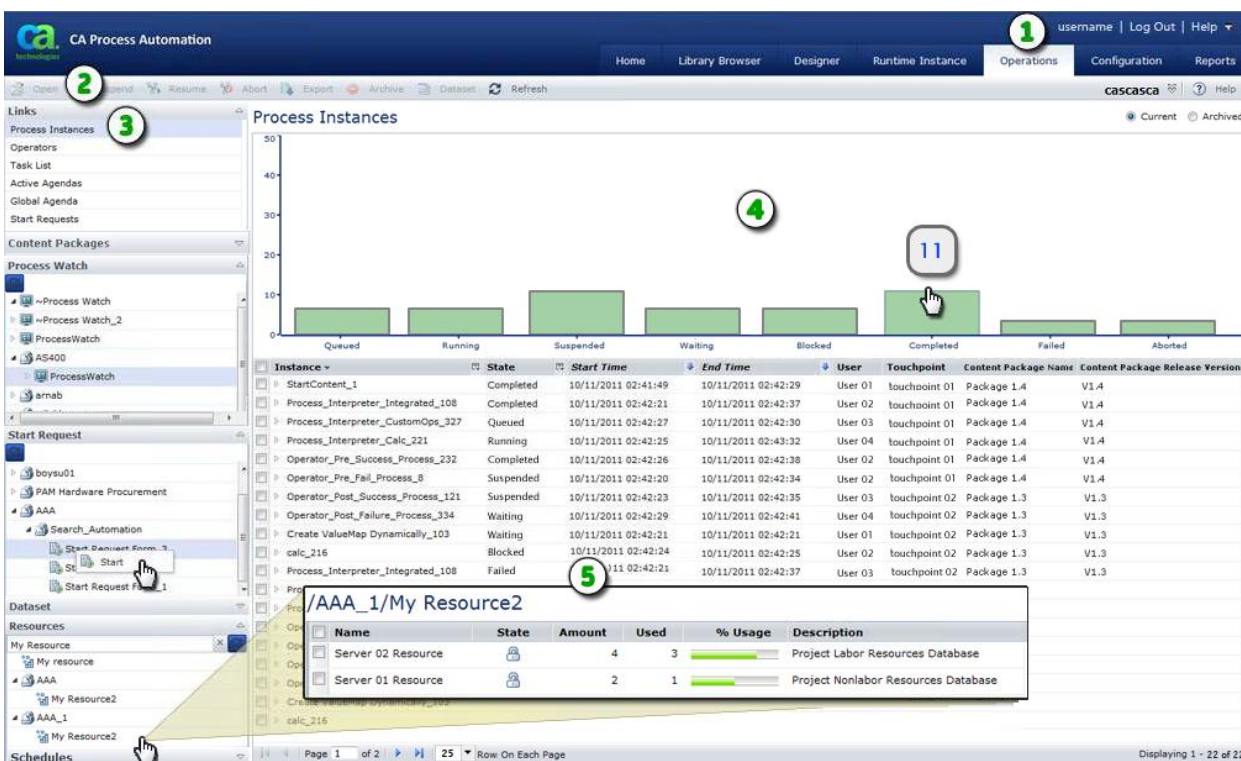
## Operações

A página Operações exibe um painel dos seguintes elementos principais:

- Links para operadores, processos e outros objetos
- Pacotes de conteúdo
- Exibição de processos
- Formulários de solicitação inicial
- Conjuntos de dados
- Recursos

A página Operações ajuda a gerenciar os processos em execução ou concluídos. Inicie os processos sob demanda e interaja com processos em execução por meio de formulários interativos. A página Operações é um painel de automação que responde aos seguintes tipos de perguntas:

- O que está sendo executado em meu ambiente?
- O que foi executado em meu ambiente?
- Quais dos processos estão disponíveis para serem iniciados sob demanda?
- Quais dos processos estão aguardando por uma entrada do usuário ou por aprovação?



**Item:**      **Descrição:**

1

**Guia Operações:** clique nessa guia para exibir a página Operações.

2

**Barra de ferramentas de Operações:** use esses comandos para trabalhar com itens nas diversas listas da página Operações.

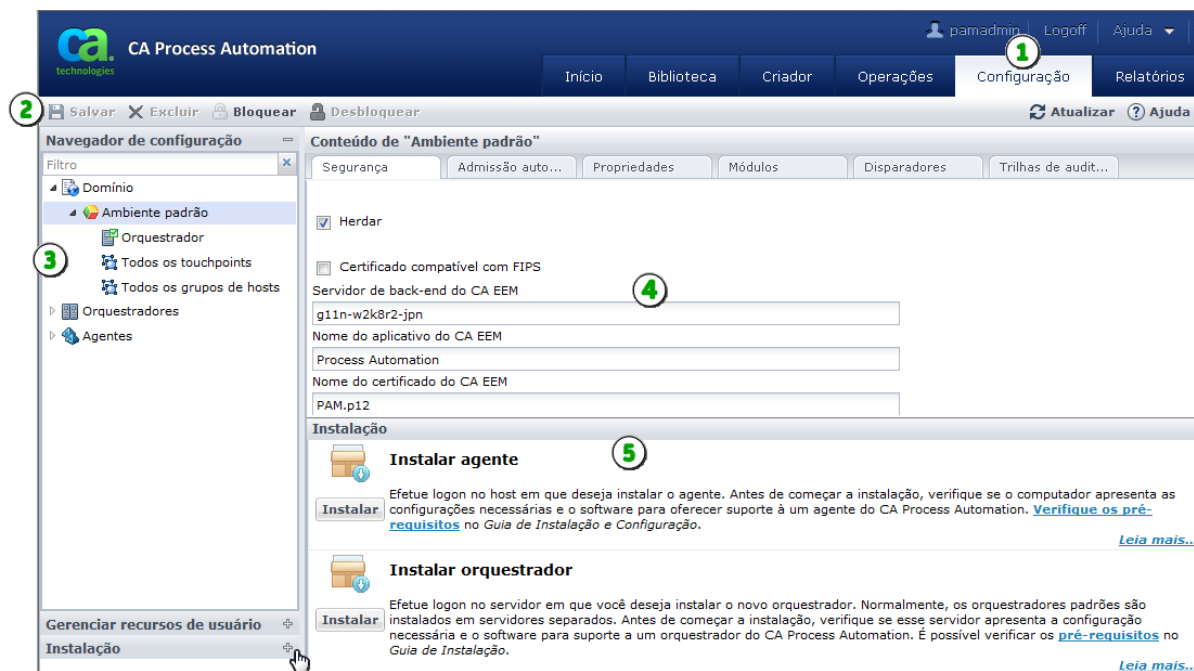
---

Item:	Descrição:
3	<p><b>Painel Operações:</b> selecione uma entrada na lista expansível neste painel. Por exemplo:</p> <ul style="list-style-type: none"><li>■ Clique em Instâncias de processo para exibir um gráfico e uma lista de instâncias.</li><li>■ Clique em Pacotes de conteúdo para exibir uma lista dos pacotes de conteúdo que foram importados, bem como seus objetos.</li><li>■ Clique em um Formulário de solicitação inicial para iniciá-lo.</li><li>■ Digite um filtro opcional para localizar um objeto de recurso.</li></ul> <p>Na ilustração, o usuário digitou Meu recurso para localizar as entradas correspondentes organizadas em pastas na seção Recursos do painel.</p>
4	<p><b>Gráfico ou Área do gráfico:</b> o produto exibirá os dados como um gráfico de instâncias de processo, exibições de processos e instâncias de processo que estão em pacotes de conteúdo. Na ilustração, o usuário aponta para a barra de instâncias Concluído para ver a contagem total (11). Clique em uma barra no gráfico para adicionar o valor de status associado ao filtro para a lista de resultados abaixo dele. Clique na barra novamente para remover o valor do filtro.</p>
5	<p><b>Página principal ou lista:</b> a área principal da página Operações exibe informações essenciais. Além do gráfico, o produto geralmente apresenta esses dados em uma tabela, lista ou formulário. Por exemplo, é possível exibir, filtrar e classificar uma lista de instâncias de acordo com seu estado. Clique em um objeto de recurso para exibir o status atual dos recursos que ele representa.</p>

---

## Configuração

A paleta Navegador de configuração exibe a hierarquia lógica do domínio. Os administradores usam a guia Configuração para instalar e configurar vários agentes e orquestradores. Os criadores de conteúdo têm acesso somente leitura à paleta Navegador de configuração na guia Configuração.



**Item:**      **Descrição:**

1

**Guia Configuração:** somente os administradores podem usar essa guia para atualizar as definições de configuração.

2

**Barra de ferramentas de configuração:** somente os administradores podem bloquear objetos no Navegador de configuração, salvar as alterações de configuração e desbloquear os objetos.

3

**Hierarquia do domínio do Navegador de configuração:** somente os administradores podem configurar o produto.

4

**Painel principal do Navegador de configuração:** o painel principal exibe os dados de configuração.

5

**Instalação e Gerenciar recursos de usuários:** somente os administradores podem exibir essas paletas. As paletas Instalação e Gerenciar recursos de usuários não são exibidas quando os criadores de conteúdo clicam na guia Configuração.

## Relatórios

Use esta página para executar tarefas de geração de relatórios. Esses recursos incluem:

- Gerar um relatório existente
- Fazer upload de um novo relatório
- Definir parâmetros personalizados
- Exportar dados de relatório

The screenshot displays the CA Process Automation web interface. At the top, there is a navigation menu with options like 'Início', 'Biblioteca', 'Criador', 'Operações', 'Configuração', and 'Relatórios'. Below the menu, there is a search bar and a 'Novo' button. The main area shows a form to 'Adicionar novo relatório' with fields for 'Nome do relatório' and 'Descrição resumida'. Below this is a table listing existing reports with columns for 'Nome', 'Descrição', 'Tipo de relatório', 'Criado por', and 'Data de modificação'. Three numbered callouts (1, 2, 3) highlight the 'Relatórios' menu item, the 'Novo' button, and the 'Adicionar novo relatório' form respectively. A fourth callout (4) points to the table of reports. Overlaid on the interface are three windows: 'Parâmetro' (5) for configuring report parameters, and two 'Visualizador de relatórios BIRT' windows (6) showing report execution results, including a line chart and a pie chart.

**Item:**      **Descrição:**

- 1 **Guia Relatórios:** clique na guia Relatórios para exibir ou criar relatórios.
- 2 **Barra de ferramentas de relatórios:** use essa barra de ferramentas para trabalhar com a página Relatórios. Uma barra de ferramentas padrão separada também está disponível para trabalhar com a saída de relatório.
- 3 **Adicionar nova área de relatório:** se você clicar em Novo na barra de ferramentas, essa área será exibida para que você possa definir e salvar detalhes do relatório.
- 4 **Lista de relatórios:** lista os relatórios disponíveis.

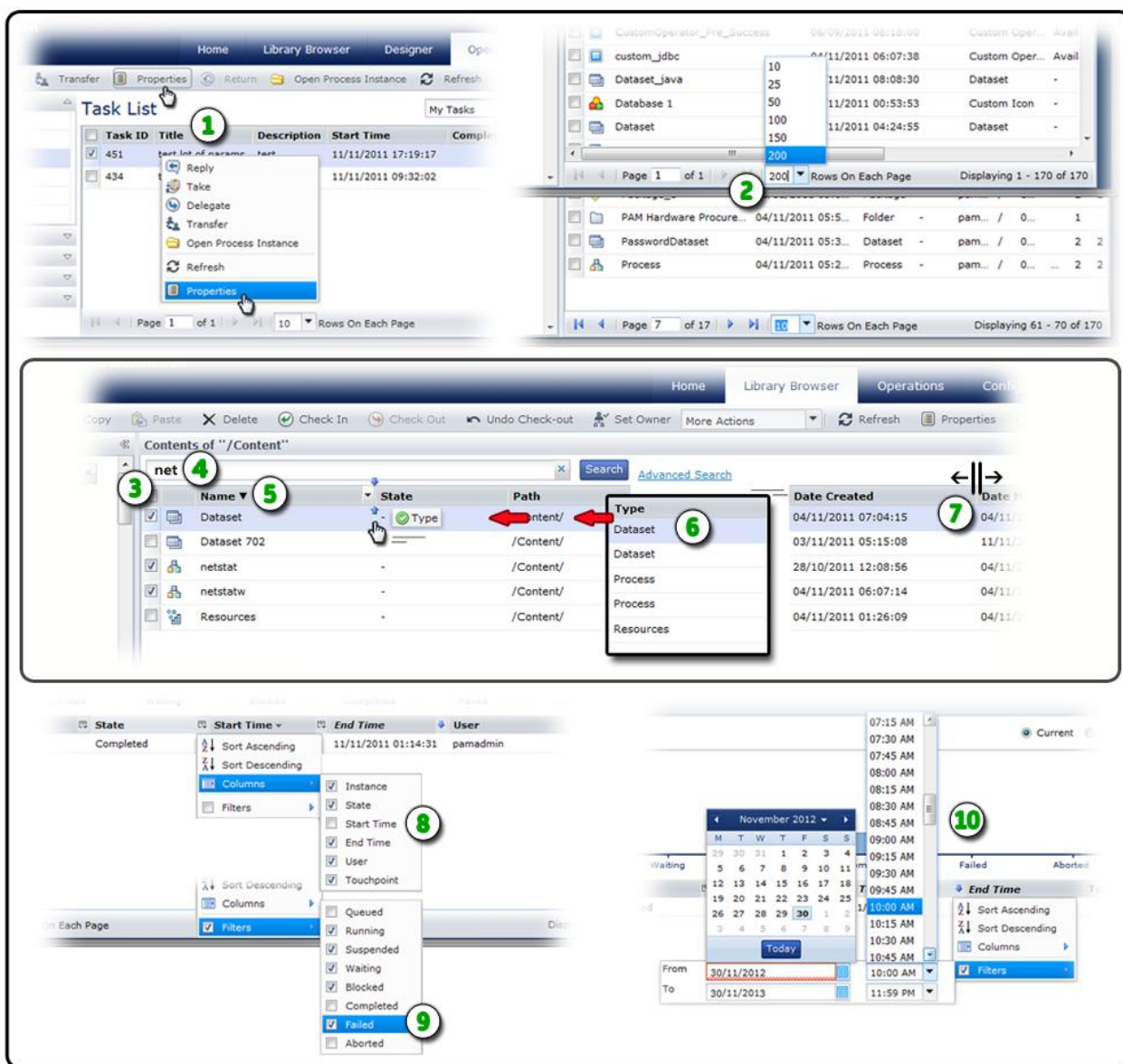
Item:	Descrição:
5	<b>Caixa de diálogo Parâmetro:</b> com exceção do Relatório de objetos de referência, quando você clica em um relatório na lista, essa caixa de diálogo é exibida. Use-a para definir o intervalo de dados a serem incluídos na instância do relatório selecionada. Neste exemplo, o usuário selecionou Todos os estados no campo {} Estado do processo.
6	<b>Saída de relatório:</b> depois de clicar em OK, uma visualização da saída na tela é exibida na caixa de diálogo Parâmetro. Você pode usar essa saída como está, imprimi-la para um outro dispositivo ou exportá-la para ser usada em outro aplicativo.

**Observação:** para criar novos relatórios, instale o criador da ferramenta BIRT em um servidor de aplicativos separado e conecte-o ao banco de dados de relatórios.

## Controles comuns de interface do usuário

A interface web do CA Process Automation oferece um conjunto consistente de controles. Tabelas, listas, páginas, comandos e filtros funcionam da mesma maneira padrão de uma página para outra.

**Observação:** alguns recursos não estão disponíveis em todas as páginas.



Item: Descrição:

---

**Item:** **Descrição:**

---


- 1 **Comandos:** como um atalho, você poderá clicar com o botão direito do mouse em um item da lista para selecionar os mesmos comandos disponíveis em uma barra de ferramentas. Neste exemplo, você pode clicar em **Propriedades** na barra de ferramentas ou clicar com o botão direito do mouse em uma linha de tarefa na tabela e escolher **Propriedades** no menu de atalho resultante.

---
- 2 **Paginação:** quando o CA Process Automation apresentar tabelas de dados, você poderá controlar quanto cabe em uma página. As tabelas superior e inferior deste exemplo exibem ambas 170 itens. Na página superior, o usuário selecionou **200** linhas em cada página. Isso condensa todos os 170 itens em uma única página. Para a página inferior, o usuário optou por exibir 10 linhas. Essa configuração aumenta o número de páginas necessárias para exibir todos os 170 itens para 17 páginas.

Use os controles na parte inferior da janela para navegar para a primeira, a última, a anterior ou a próxima página na tela. Você também pode saltar para uma página específica, inserindo o número de página no campo Página.

---
- 3 **Seleção múltipla:** marque uma ou mais linhas para aplicar o mesmo comando a itens semelhantes. O aplicativo só permitirá que você execute comandos válidos para vários itens ao mesmo tempo.

---
- 4 **Filtros:** digite uma palavra ou uma sequência de caracteres de texto opcional e pressione a tecla Enter para exibir somente os itens correspondentes. Por exemplo, digite *net* e somente os objetos *netstat* e *netstatw* aparecerão na lista. Clique no X para desmarcar o filtro.

---
- 5 **Ordem de classificação:** clique em  em uma linha de cabeçalho de coluna de campo para selecionar **Classificação crescente** ou **Classificação decrescente**. Da mesma forma que em um atalho, clique no cabeçalho da coluna para alternar a ordem de classificação. Um pequeno triângulo indica a direção da ordem de classificação: crescente (para baixo), decrescente (para cima) ou sem classificação (sem triângulo).

---
- 6 **Organização da coluna:** clique e arraste uma coluna inteira para alterar a sequência das colunas em uma tabela. Neste exemplo, o usuário optou por posicionar a coluna Tipo entre Nome e Estado.

---
- 7 **Tamanho de coluna:** posicione o mouse sobre a linha divisória que divide as duas colunas. Quando ele for alterado para um cursor de redimensionamento, arraste a borda esquerda ou direita da coluna para ajustar a largura da coluna e seus dados.

---
- 8 **Mostrar ou ocultar colunas:** você pode mostrar (marcar) ou ocultar (desmarcar) os campos listados para controlar as colunas que aparecerão em uma tabela.

---
- 9 **Filtros rápidos:** se estiver disponível, você poderá aplicar um filtro rápido a uma coluna. Verifique os valores de campo que deseja incluir e desmarque os valores que deseja excluir.

---

---

**Item: Descrição:**

---



**Filtros de data e hora:** para os campos de data e hora de uma tabela, você pode definir um filtro como um intervalo ou um período de tempo. O filtro exclui os dados que estiverem fora do intervalo. Neste exemplo, o usuário deseja incluir apenas as linhas com valores de Hora de término desde 30 de novembro de 2012, às 10:00, até 30 de novembro de 2013, às 23:59.

---

## Explorar o conteúdo pronto para uso

O link Conteúdo pronto para uso na página Início liga o criador de conteúdo ao conteúdo de exemplo. O conteúdo de exemplo ilustra o que é possível fazer com vários objetos de automação.

### Siga estas etapas:

1. Clique na guia Início e, em seguida, clique em Explorar o conteúdo pronto para uso.  
A guia Biblioteca é aberta com a pasta PAM\_PreDefinedContent selecionada no painel de navegação.
2. Expanda a pasta chamada PAM\_PreDefined Content.
3. Expanda a subpasta que representa o operador, o objeto ou o processo de criação de interesse.
4. No painel principal, clique duas vezes em um objeto em que o Tipo é Processo.  
O processo é exibido na guia Criador. O processo contém anotações que descrevem a finalidade do processo, outras informações relevantes e instruções para personalização do processo.
5. Use o processo de exemplo selecionado como uma ferramenta de aprendizado ou como um modelo para um processo próprio.

## Usar o CA EEM para alterar sua senha do CA Process Automation

O administrador geralmente atribui uma senha temporária ao configurar contas de usuário para o armazenamento de usuários interno. Todos os usuários do CA Process Automation com contas de usuário criadas no CA EEM podem alterar essa senha antes de efetuar logon no CA Process Automation. Em seguida, você poderá alterar sua senha do CA Process Automation no intervalo definido pelas diretivas de senha.

**Observação:** esse recurso não se aplica quando o CA EEM fizer referência a contas de usuários de um armazenamento de usuários externo, como o Microsoft Active Directory. Nesse caso, mantenha sua senha no diretório referenciado.

Use o CA EEM para alterar sua senha do CA Process Automation.

### Siga estas etapas:

1. Abra um navegador e digite o URL para o servidor do CA EEM que o CA Process Automation utiliza. Por exemplo:  
**https://nome\_do\_host\_ou\_endereço\_IP:5250/spin/eiam/**  
Para identificar o nome do host ou o endereço IP do CA EEM que o CA Process Automation utiliza, consulte o campo Servidor de back-end do CA EEM, na guia Configuração e na subguia Segurança do CA Process Automation.
2. Faça logon no CA Embedded Entitlements Manager (CA EEM) na caixa de diálogo Efetuar logon:
  - a. Para Aplicativo, selecione <Global>.
  - b. Exclua EiamAdmin se esse nome padrão for exibido no campo Nome do usuário.
  - c. Digite seu nome de usuário e senha do CA Process Automation.
  - d. Clique em Efetuar logon.
3. Em Autoadministração, clique em Alterar senha.
4. Redefinir sua senha:
  - a. Digite seu nome de usuário do CA Process Automation e a senha antiga.
  - b. Em seguida, digite a nova senha nos campos Nova senha e Confirmar senha.
  - c. Clique em OK.

O CA Process Automation aceita as credenciais atualizadas quando você efetua logon.

## Navegadores da web

Mesmo entre navegadores da web suportados, o CA Process Automation pode se comportar de modo diferente devido a diferenças nos produtos de terceiros. Por exemplo, quando você tenta manter uma página ou guia com alterações não salvas, cada um dos seguintes navegadores com suporte pode exibir uma mensagem um pouco diferente:

- Microsoft Internet Explorer
- Google Chrome
- Mozilla Firefox

Essencialmente, todos os três fornecem a opção de permanecer na página para salvar os dados ou sair da página e descartar as alterações.

### Microsoft Internet Explorer

Quando você configura o Microsoft Internet Explorer 8 ou 9 para CM (Compatibility Mode - Modo de Compatibilidade), o CA Process Automation pode apresentar um desempenho ou comportamento ruim. Por exemplo, os menus suspensos para as colunas dos campos na página Operações podem desaparecer se você clicar em Solicitações iniciais (ou outra opção no painel Operações) e, em seguida, voltar para Links e selecionar Instâncias de processo. Uma vez nesse estado, o botão do menu suspenso do cabeçalho da coluna pode não ser exibido para outras exibições. Isso pode resultar na incapacidade de definir um filtro.

Recomendamos que você não ative explicitamente o CM ou Configurações da Intranet. Se você observar um comportamento estranho ao usar o Internet Explorer, certifique-se de que o CM não está ativado.

**Observação:** se você precisar ativar Configurações da Intranet, desative manualmente a opção *Exibir sites da intranet na exibição de compatibilidade* em Configurações do Modo de Exibição de Compatibilidade no menu Ferramentas do Internet Explorer.



# Capítulo 3: Navegador da biblioteca

---

Esta seção contém os seguintes tópicos:

[Personalizar o Navegador da biblioteca](#) (na página 59)



[Pesquisar o Navegador da biblioteca](#) (na página 59)

[Trabalhando com objetos e pastas](#) (na página 64)

## Personalizar o Navegador da biblioteca

É possível personalizar a exibição das colunas no Navegador da biblioteca. As colunas exibem os valores de campo para as entradas de uma tabela.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. No painel de pastas do Navegador da biblioteca, selecione uma pasta.
3. Na janela principal, para mostrar ou ocultar uma coluna, clique em  no cabeçalho da coluna e escolha Colunas.  
Uma lista de verificação de nomes de campo disponíveis é exibida.
4. Desmarque as colunas dos campos que você deseja ocultar. Marque as colunas dos campos que você deseja exibir.
5. Para classificar a lista, clique em  em um cabeçalho de coluna específico e escolha Classificação crescente ou Classificação decrescente.
6. Para reorganizar as colunas, clique no cabeçalho de uma coluna e arraste-o para a esquerda ou direita.

## Pesquisar o Navegador da biblioteca

Use os recursos de pesquisa para localizar itens específicos no Navegador da biblioteca. Em seguida, você poderá executar ações em vários objetos relacionados nos resultados da pesquisa.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. No painel de pastas do Navegador da biblioteca, selecione a pasta raiz ou uma subpasta específica.

O produto limita a pesquisa aos objetos e pastas na pasta selecionada.

3. No campo Pesquisar, digite o nome de um operador ou de uma pasta.
4. (Opcional) Clique em Pesquisa avançada para basear uma pesquisa refinada em qualquer uma das seguintes propriedades:
  - Nome
  - Proprietário
  - Tipo
  - Estado Disponibilizar
  - Data de criação
  - Data de modificação
  - Tags

5. Clique em Pesquisar.

Os resultados da pesquisa são exibidos.

**Observação:** se você pesquisar em uma hierarquia de pastas com um pacote de conteúdo, o produto retornará apenas os objetos na versão da release atual do pacote de conteúdo. O produto também retorna objetos armazenados na hierarquia de pastas, mas que não estão incluídos no pacote de conteúdo.

6. Trabalhe com objetos e pastas nos resultados da pesquisa como faria em qualquer outra pasta da biblioteca.
  - Para abrir um item para edição, clique duas vezes nele.
  - Para exibir um menu de atalho de comandos disponíveis, clique com o botão direito do mouse em um único item.
  - Para aplicar uma ação a vários objetos ao mesmo tempo, selecione os itens e clique com o botão direito do mouse em um item selecionado.
7. Para iniciar uma nova pesquisa, execute uma das ações a seguir:
  - Clique em Redefinir.
  - Clique em Pesquisa básica ou Pesquisa avançada.
  - Limpe ou modifique os critérios especificados nos campos de pesquisa.
  - Clique no x da caixa Pesquisa básica.

## Pesquisar por informações da versão da release

Também é possível associar cada versão do objeto com uma versão da release. A guia Versões usa suas associações exibidas para filtrar os objetos e as versões que estão associadas a uma sequência de caracteres que você especificar. É possível procurar por uma determinada versão da release para exibir sua versão de interesse. Como alternativa, é possível pesquisar a sequência de caracteres genérica que retorna várias versões do objeto. A guia Versões oferece suporte a pesquisas com ou sem caracteres curinga (\*), como mostra o exemplo a seguir:

Sequência de caracteres de pesquisa	Resultados da pesquisa
5	
*5	Qualquer versão do objeto no qual o campo Versão ou o campo Versão da release contém o caractere 5.
5*	
*5*	

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse em um objeto e, em seguida, selecione Editar.
3. No editor de objetos, clique na guia Versões.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campo e exemplos.

4. Na barra de ferramentas de Versão, clique no ícone de lupa no lado mais à direita.



5. Especifique critérios de pesquisa no campo de pesquisa e, em seguida, clique em Pesquisar.

A pesquisa retorna os objetos e as versões que estão associadas à sequência de caracteres especificada.

## Pesquisar por informações da versão

A guia Versão permite que você filtre as versões exibidas para objetos que contenham a sequência de caracteres especificada.

- Para os critérios de pesquisa diferentes de datas, os resultados da pesquisa incluem linhas para versões do objeto que contenham a sequência de caracteres especificada em qualquer um dos seguintes campos:
  - Versão
  - Modificado por
  - Criado por
- **Observação:** para exibir os objetos que contenham 5, por exemplo, especifique 5, \* 5, 5 \*, ou \* 5 \* como critérios de pesquisa.
- Para critérios de pesquisa de data, os resultados da pesquisa incluem linhas para versões do objeto que tenham a data inserida em um dos seguintes campos:
  - Criado em
  - Última modificação em

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse em um objeto e selecione Editar.  
O editor de objetos é exibido.
3. Clique na guia Versões.
4. Clique no ícone de lupa, do lado direito da barra de ferramentas Versão.  
O campo de pesquisa e o botão Pesquisar são exibidos.
5. Especifique critérios de pesquisa no campo de pesquisa e clique em Pesquisar.

O processo de pesquisa exibe os objetos que atendem aos critérios de pesquisa especificados. Critérios de pesquisa diferentes de datas retorna todas as versões que contenham a sequência de caracteres especificada nos campos Versão, Modificado por e Criado por. Critérios de pesquisa de datas retornam todas as versões com uma correspondência exata nos campos Criado em e Última modificação em.

## Pesquisar por informações de trilha de auditoria

A guia Trilha de auditoria permite que você exiba somente as versões de objetos que atendam a um ou aos dois seguintes critérios de pesquisa:

- O campo Versão ou o campo Nome de usuário contém a sequência de caracteres especificada no campo Filtro.  
Por exemplo, se for especificado 5 no campo Filtro, a pesquisa retornará qualquer objeto Trilha de auditoria que tenha 5 no campo Versão ou 5 no campo Nome de usuário
- O campo Última atualização contém uma data no intervalo de data especificado nos campos De e Até.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse em um objeto e selecione Editar.  
O editor de objetos é exibido.
3. Clique na guia Trilhas de auditoria.
4. Clique no ícone de lupa, do lado direito da barra de ferramentas Trilha de auditoria.  
Os campos de pesquisa e o botão Pesquisar são exibidos.
5. Digite os critérios de pesquisa no campo Filtro, os campos De e Até, ou ambos. Em seguida, clique em Pesquisar.

### Filtrar

Especifica os critérios de pesquisa numérico ou alfanumérico para os campos a seguir, em que os resultados da pesquisa incluem quaisquer versões do objeto que contenham a sequência de caracteres especificada.

- Versão
- Nome de usuário

### De - Até

Especifica um intervalo de datas para pesquisar os dados da Última atualização.

O processo de pesquisa exibe os objetos que atendem aos critérios de pesquisa especificados.

## Trabalhando com objetos e pastas

As pastas são recipientes para objetos.

Use este processo para editar um objeto:

1. Identifique o local de um objeto ou pasta no Navegador da biblioteca.
2. Abra o objeto.
3. Reserve o objeto para uma versão de trabalho que possa ser editada.
4. Edite o objeto.
5. Salve as mudanças.
6. Disponibilize o objeto (nessa etapa, você também decide se aplicará as alterações à mesma versão ou a uma nova versão do objeto).

Se desejar apenas inspecionar um objeto, clique em Propriedades ou abra-o. Dessa maneira, você pode visualizar suas propriedades em detalhes e decidir se deseja editá-lo.

Os objetos são abertos em um editor ou criador que se baseia no tipo de objeto. Por exemplo, um processo é aberto no Criador de processos e um objeto do conjunto de dados é aberto em uma caixa de diálogo Conjunto de dados. Cada editor tem guias, paletas e botões da barra de ferramentas específicos.

**Observação:** esta seção inclui apenas informações básicas sobre como abrir e editar objetos e gerenciar versões. Outras seções descrevem, com mais detalhes, como editar vários tipos de objetos.

## Tipos de objetos de automação

Os objetos de automação são componentes de aplicativo que definem os elementos configuráveis de um pacote do CA Process Automation. Esses objetos definem as operações do sistema e incluem um software executável. Crie e configure objetos de automação do CA Process Automation no navegador da biblioteca, em pastas de biblioteca de automação específicas associadas a um determinado orquestrador.

Você pode executar as seguintes ações em uma biblioteca de automação:

- Explorar uma biblioteca de automação associada a um orquestrador.
- Criar, editar e visualizar objetos em uma biblioteca de automação.
- Criar pastas em uma biblioteca para agrupar objetos relacionados. As pastas permitem que você defina uma estrutura hierárquica, de modo que você e seus colegas de trabalho possam localizar os objetos. Essa estrutura é semelhante aos diretórios ou às pastas de sistema operacional do computador.

É possível criar os seguintes tipos de objeto de automação no navegador da biblioteca:

#### **Tipo**

Um objeto de *processo* mostra graficamente a ordem e as dependências entre os operadores e alguns outros processos. Processos representam graficamente os operadores com links que mostram a sequência e a lógica por trás das etapas que o processo realiza.

#### **Programação**

Um objeto de *programação* aplica condições de data e hora para quando os módulos (inclusive processos) forem executados. Para agrupar tarefas por aplicativo, propriedade ou outros critérios, use várias programações.

#### **Calendário**

Um objeto de *calendário* define as regras que descrevem condições de data complexas. Um objeto de calendário especifica graficamente datas, intervalos de tempo e elementos condicionais que determinam quando e com que frequência o produto executa operações.

#### **Ícone personalizado**

Um objeto de *ícone personalizado* especifica imagens gráficas que identificam os operadores de maneira exclusiva.

#### **Operador personalizado**

Um objeto de *operador personalizado* permite que você estenda a apresentação e a configuração de operadores existentes. Você pode basear um operador personalizado em um operador existente e otimizá-lo com parâmetros específicos criados para serem reutilizados em vários processos.

#### **Conjunto de dados**

Um objeto de *conjunto de dados* define e agrupa variáveis que o produto usa como parâmetros que outros processos, operadores e recursos exigem. Locais de aplicativos, senhas e nomes de perfil são alguns exemplos. Você pode facilmente configurar essas variáveis, de modo que o produto possa atualizar os processos e programá-los de forma eficiente para refletir as alterações em um ambiente de aplicativo.

#### **Pacote**

A partir do CA Process Automation 04.2.00, o objeto de *pacote* não está mais disponível para agrupar atalhos para outros objetos de automação. Os pacotes não estão disponíveis para referência por meio de nenhum dos navegadores de objetos do CA Process Automation. Para exportar e, em seguida, importar conteúdo não modificável de um ambiente para outro na release 04.2.00, use um pacote de conteúdo.

### Pacote de conteúdo

O objeto de automação de *pacote de conteúdo* agrupa os objetos do CA Process Automation que um criador de conteúdo exportou como pacote de conteúdo. O criador de conteúdo adiciona os objetos de automação em uma pasta e exporta a pasta como pacote de conteúdo. O usuário de produção pode visualizar os pacotes de conteúdo importados na biblioteca ou na guia Operações.

### Exibição de processos

Um objeto de *exibição de processos* permite que os usuários definam e monitorem aplicativos selecionados no ambiente de produção. Um objeto Exibição de processos é um conjunto de atalhos para outros elementos da biblioteca de automação. Um usuário pode abrir um objeto Exibição de processos para exibir o estado das instâncias do processo e outros objetos. Um objeto de exibição de processos permite que um usuário monitore os operadores sem necessariamente permitir acesso a objetos ou dados subjacentes.

### Recurso

Um objeto de *recurso* é um modelo que representa elementos da arquitetura de seu sistema. Use-o para sincronizar processos independentes que dependem de elementos comuns da infraestrutura e para quantificar e controlar o acesso a entidades de TI específicas. Inclua vários recursos que representem entidades relacionadas em um único objeto Recurso.

### Formulário de solicitação inicial

Um objeto de *formulário de solicitação inicial* define atalhos para permitir que um usuário de produção inicie processos manualmente. Uma caixa de diálogo personalizada solicita aos usuários os valores de parâmetros necessários para iniciar seus processos associados.

### Formulário de solicitação de interação

Um objeto de *formulário de solicitação de interação* permite que você solicite que os usuários forneçam respostas em campos de dados e outros controles de interface de usuário. Os usuários inserem as informações exigidas para continuar um processo. Por exemplo, use um formulário de solicitação de interação para solicitar que uma das partes interessadas revise cada etapa de um processo de aprovação.

## Criar uma pasta

Hierarquias de pastas permitem categorizar e organizar objetos relacionados e permitem que você aplique recursivamente as permissões selecionadas a objetos e pastas subordinados. Para organizar e proteger grupos de objetos, é possível adicionar pastas para estruturar a hierarquia da biblioteca.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. No painel Navegador da biblioteca, selecione a pasta na qual deseja criar uma nova pasta.
3. Na barra de ferramentas, clique em Novo e, em seguida, escolha Pasta.  
A nova pasta é exibida.
4. Renomear a pasta.

**Observação:** para definir direitos e executar outras tarefas de objeto, a prática recomendada é organizar os objetos em pastas. Não crie objetos no nível raiz, pois não é possível gerenciá-los como um grupo.

## Criar um objeto

Crie novos objetos de automação em pastas específicas da biblioteca que está associada a um orquestrador. Depois de criar um objeto, é possível editar suas propriedades.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. No painel de pastas Navegador da biblioteca, selecione uma pasta para armazenar o novo objeto.
3. Na barra de ferramentas, clique em Novo e selecione um tipo de objeto.  
Um novo objeto é criado.
4. Forneça um nome exclusivo para o objeto.  
O aplicativo o alerta quando o nome especificado já estiver associado a outro objeto na pasta.

**Observação:** para definir direitos e executar outras tarefas de objeto, a prática recomendada é organizar os objetos em pastas. Não crie objetos no nível raiz, pois não é possível gerenciá-los como um grupo.

**Observação:** o objeto de pacote não é mais suportado no CA Process Automation 4.2.00. Agora, um criador de conteúdo pode agrupar os objetos de automação em uma pasta e exportar objetos como pacote de conteúdo para a implantação em um ambiente de produção. O uso do objeto de pacote existente é mantido para compatibilidade com versões anteriores. Consulte Como migrar os objetos de um pacote para uma nova pasta para se preparar para a exportação.

## Editar um objeto

Após criar um objeto, você poderá modificar sua configuração a qualquer momento. Por exemplo, você pode:

- Editar as tags de um objeto
- Adicionar um recurso a um objeto de recurso
- Remover um objeto de uma exibição de processo.

O CA Process Automation armazena várias versões de um objeto. Por padrão, quando você reserva e edita um objeto, o aplicativo usa a versão *atual* designada. Depois de editá-lo, disponibilize o objeto. É possível especificar se as alterações devem ser aplicadas à mesma versão ou a uma nova versão quando você disponibilizar o objeto. Em qualquer um dos casos, o aplicativo aplicará as alterações por padrão.

Cada objeto pode ter uma [versão da release](#) (na página 454) associada. A versão da release identifica a versão específica de um objeto para implantação em um ambiente de produção. O CA Process Automation exige que o objeto seja entregue para o ambiente de produção em um [objeto de pacote de conteúdo](#) (na página 463). É possível definir um sinalizador que impeça que a versão da release seja modificada e, em seguida, o CA Process Automation define o objeto como versão base na importação para o ambiente de produção. Quando uma versão da release é definida para um objeto, a versão do objeto interno fica associada a esse número da release.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. No painel de pastas do Navegador da biblioteca, clique em uma pasta.
3. Clique duas vezes em um objeto.  
A versão atual do objeto é aberta.
4. Se o objeto ainda não estiver reservado, clique em Reservar.
5. Edite o objeto.
6. Clique em Salvar ou em Salvar e fechar.  
O aplicativo salva suas alterações. Como o objeto permanece reservado, é possível continuar a editá-lo.
7. Clique em Disponibilizar.
8. Realize o seguinte procedimento na caixa de diálogo Disponibilizar:
  - a. No campo Disponibilizar como, selecione Nova versão ou Mesma versão.
  - b. (Opcional) Insira comentários sobre as alterações feitas nesta versão.

- c. (Opcional) Marque a caixa de seleção Linha de base para usar esta versão como uma linha de base.
- d. (Opcional) Desmarque a caixa de seleção Atual se desejar uma nova versão, mas não quiser marcá-la como atual. A versão atual anterior permanece a atual.
- e. Clique em Disponibilizar.

## Exibir ou alterar as propriedades gerais de um objeto

É possível visualizar as propriedades gerais de qualquer objeto ou processo da biblioteca.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Selecione uma pasta, clique com o botão direito do mouse em um objeto e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia Geral.
4. Visualize as propriedades do objeto somente leitura.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter as descrições dos campos.

5. (Opcional) Edite o campo Descrição de um objeto reservado ou disponibilizado. Clique em Salvar para armazenar as alterações.

## Especificar tags ou palavras-chave para objetos

A guia Tags permite atribuir palavras-chave que organizem objetos de acordo com qualquer sistema de nomenclatura significativo reconhecido por sua organização. Dessa maneira, você e outros usuários poderão executar uma pesquisa avançada de objetos ou pastas usando as tags ou as palavras-chave.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Selecione uma pasta, clique com o botão direito do mouse em um objeto e, em seguida, selecione Propriedades.

3. No painel Propriedades, clique na guia Tags.
4. Na guia Tags:
  - a. Clique em Editar.
  - b. Digite um ou mais valores de texto separados por vírgula e sem espaços entre os itens da lista. Por exemplo:  
`testing, security, Project Code Beta`
  - c. Clique em Salvar.

### Alterar a propriedade para a automação de objetos

O usuário que cria um objeto de automação ou pasta é, por padrão, o proprietário. O proprietário tem controle total sobre o objeto de automação ou a pasta. Um proprietário pode alternar a propriedade para outro usuário do CA Process Automation.

**Observação:** a permissão `Environment_Content_Administrator` do CA EEM concede controle total sobre todos os objetos de automação e pastas. Todos os administradores que pertencem ao grupo `PAMAdmins` têm esta permissão.

Se você ativar a segurança em tempo de execução, apenas o proprietário do processo (ou um administrador) poderá iniciar esse processo.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione um ou mais objetos, incluindo as pastas.
3. Clique no botão Definir proprietário, na barra de ferramentas.
4. Na lista Usuários disponíveis, selecione a conta de usuário a ser definida como o novo proprietário. Use a pesquisa para encontrar contas de usuário correspondentes.
5. Clique em Salvar e fechar.

## Especifique uma diretiva de arquivamento

Depois que um processo automatizado for executado e concluído com êxito, o produto arquivará os dados de tempo de execução para referência e métricas. Por padrão, o objeto herda as propriedades Diretiva de arquivamento do orquestrador. No entanto, é possível definir manualmente o número de dias em que o servidor mantém objetos de instâncias de processo ou de programação concluídos ou com falha. Você também pode definir o número de instâncias de processo concluídas ou com falha que o produto manterá.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Selecione uma pasta e, em seguida, selecione um objeto de programação ou processo.
3. Clique com o botão direito do mouse no objeto e, em seguida, selecione Propriedades.
4. No painel Propriedades, clique na guia Diretiva de arquivamento.
5. Desmarque a caixa de seleção Herdar diretiva de arquivamento do orquestrador. Esta tarefa ativa os outros campos.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campos e exemplos.

6. Digite um valor para histórico de processos.

Esse valor define o intervalo (em dias) em que o produto manterá instâncias de processo concluídas e com falha no arquivo antes de limpar automaticamente os registros. O produto mantém o histórico de processos para esse intervalo independentemente do número de instâncias no arquivo.

7. Digite os valores de instâncias com falha e concluídas.

Esses valores definem o número de instâncias de processo que o produto manterá antes de eliminar automaticamente os registros com os seguintes status:

- Apenas status Com falha
- Status Concluído

Nas instâncias que excederem o limite de retenção especificado, o produto excluirá primeiro os registros mais antigos.

8. Clique em Salvar.

## Especificar a release da pasta

Um criador de conteúdo pode fornecer informações da release para uma pasta. A guia Release nas propriedades da pasta fornece a versão da release da pasta. A especificação da versão da release de uma pasta é necessária para exportar a pasta como pacote de conteúdo. A adição de informações que descrevem o conteúdo da pasta pode ajudar o consumidor do pacote de conteúdo quando o conteúdo for importado para um novo ambiente.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse no painel direito da biblioteca e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia Release para exibir as propriedades da pasta.
4. Clique no campo Valor do atributo ReleaseVersion.
5. Na janela Valor, digite a Versão da release da pasta e, em seguida, clique em OK.

**Observação:** não é possível excluir ou editar o atributo ReleaseVersion, mas é possível inserir a versão da release apropriada na coluna Valor.

6. Clique em Salvar.
7. Clique em Adicionar propriedade ou Remove Property para definir ou remover mais propriedades da pasta.

Por exemplo, você pode adicionar as informações a seguir na guia Release:

- Proprietário que criou o pacote de conteúdo.
- Usuários que podem usar os objetos do pacote de conteúdo após a importação para um novo ambiente.
- Informações de suporte.
- Ajuda online (URL da Ajuda online).

## Especificar propriedades de ROI

Quando você definir as propriedades de ROI para um objeto de processo, os valores que você digitar serão exibidos no conjunto de dados de instâncias de processo.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Selecione uma pasta, clique com o botão direito do mouse em um objeto de processo e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia ROI.
4. Marque a caixa de seleção Ativar ROI.

5. Digite os seguintes valores:
  - No campo Tempo gasto com mão-de-obra manual, digite o tempo da equipe total (em horas e minutos) necessário para disparar várias operações de processo manualmente.
  - No campo Tempo gasto com processamento manual, digite o tempo da equipe total (em horas e minutos) necessário para executar todo o processo manualmente.
  - Na lista suspensa Criticidade, selecione a criticidade do processo que aparece no relatório de ROI (alta, média ou baixa).
  - (Opcional) No campo Nome do grupo de aplicativos/sistema, digite o nome do grupo comum por meio do qual o produto agrega processos no relatório de ROI.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campos e exemplos.

6. Clique em Salvar.

## Determinar quando selecionar Executar como proprietário

O CA Process Automation fornece um controle de acesso refinado de operações em objetos de automação específicos, como processos, conjuntos de dados e programações. O controle de acesso não inclui apenas os direitos tradicionais de leitura e gravação, mas também direitos de iniciar um processo e monitorar suas instâncias. O acesso a outros objetos, como processos filho, conjuntos de dados ou calendários também é controlado. Os direitos de acesso são aplicados em todas as interfaces externas, incluindo serviços web e interface de usuário.

Considere o seguinte exemplo com essas suposições:

- O Processo P1 tem um Processo filho P2.
- O Proprietário X possui o P1 e o P2.
- O P1 exige acesso a outros elementos que o Proprietário X não deseja expor diretamente para outros usuários.
- O acesso padrão é que um processo seja executado como o usuário que iniciou o processo.

Se o Usuário A iniciar uma instância do P1, o P1 poderá iniciar o P2 apenas se o Usuário A estiver autorizado a iniciar o P2 diretamente. Para proteger os objetos de automação da exposição, o Proprietário X pode exigir que o Processo P1 (ou qualquer um de seus processos filho ou operadores) seja executado como proprietário, isto é, seja executado sob a identidade do Proprietário X. Quando o Proprietário X configura um processo para Executar como proprietário, está autorizando outros usuários a executarem o Processo P1 sem conceder a eles acesso aos elementos que o P1 precisa.

De maneira semelhante, o Proprietário X pode forçar os operadores a serem executados sob a identidade do usuário que iniciou a instância do processo. Embora o Processo P1 esteja configurado para Executar como proprietário, o Proprietário X pode forçar o Processo filho P2 a ser executado sob a identidade do usuário que iniciou direta ou indiretamente o Processo P1. Nesse caso, o CA Process Automation usa a identidade do usuário que o originou para validar o acesso ao Processo filho P2.

Considere este cenário: um usuário DBA cria um processo de administração de banco de dados que exige a exteriorização das credenciais DBA para um conjunto de dados nomeado. O DBA configura esse processo para ser executado como proprietário. Executar como proprietário concede às instâncias do processo acesso ao conjunto de dados que armazena as credenciais DBA. O DBA autoriza os usuários selecionados a executar um processo DBA específico sem conceder a esses usuários acesso direto às credenciais DBA.

Em resumo, uma instância de processo em execução possui duas identidades relacionadas:

- O usuário real, que inicia a instância do processo a partir da interface de usuário, de uma agenda ou de um serviço web.
- O usuário efetivo, que é determinado pela maneira como Executar como proprietário será aplicado às definições do processo e aos operadores nesses processos.

**Observação:** a identidade do usuário efetivo e atual é usada para validar o acesso aos processos filho, operadores e conjuntos de dados.

**Mais informações:**

[Diretrizes para definir a Segurança de tempo de execução para um processo](#) (na página 430)

## Especificar propriedades de segurança em tempo de execução

Você pode especificar as propriedades de segurança em tempo de execução para objetos Processo e Programação. Essas propriedades estabelecem como e se o sistema verifica as permissões de usuário quando os usuários tentarem acessar objetos de automação do banco de dados.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione uma pasta, clique com o botão direito do mouse em um objeto de Processo ou Programação e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia Segurança em tempo de execução.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter as descrições dos campos.

4. Em Segurança em tempo de execução, execute uma das seguintes tarefas:
  - Herde configurações de segurança ativadas do orquestrador, se aplicável. Esse valor é o padrão.
  - Selecione **Ativar** na lista suspensa. Essa opção permite que o produto verifique as permissões de usuário ao acessar objetos no processo.

Os objetos de processo geralmente são projetados para acessar outros objetos para leitura ou execução enquanto são executados, como, por exemplo, quando um processo dispara outro processo. Se ativada, o produto usa a segurança em tempo de execução para verificar se o usuário (proprietário ou chamador) tem permissões para acessar o objeto. Por exemplo, o Usuário A tem permissões para iniciar um processo. Esse processo tenta acessar o Objeto B e o Objeto C. O produto verifica as permissões para cada objeto. Se a segurança em tempo de execução determinar que o usuário não possui permissões para o Objeto B, ocorrerá uma falha no processo.
  - Selecione **Desativar** para desativar a validação e a aplicação da propriedade do processo em tempo de execução. Esta opção oferece compatibilidade dos processos existentes com as versões anteriores e permite que esses processos sejam executados como antes dessa release.
5. Conclua a configuração de segurança em tempo de execução para objetos Processo com as seguintes opções. A manutenção da caixa de seleção Executar como proprietário exige que você seja o proprietário do processo ou um Administrador de conteúdo do ambiente.
  - Execute o processo como o usuário que iniciou a instância do processo.
    - a. No campo Segurança em tempo de execução, clique em Ativar.
    - b. Desmarque Executar como proprietário.

Esta opção aplica direitos de segurança em tempo de execução com a identidade do usuário que inicia a instância do processo.
  - Execute o processo como o proprietário.
    - a. No campo Segurança em tempo de execução, clique em Ativar.
    - b. Selecione Executar como proprietário.

Esta opção aplica direitos de segurança em tempo de execução com a identidade do proprietário, independentemente de quem iniciar a instância do processo. Isso permite que os usuários iniciem um processo que terá acesso aos elementos que o proprietário deseja proteger.
  - Marque a caixa de seleção Ativar a recuperação do operador para a recuperação automática de operadores específicos que falham com um SYSTEM\_ERROR.
6. Clique em Salvar.

**Mais informações:**

[Diretrizes para definir a Segurança de tempo de execução para um processo](#) (na página 430)

## Definir a duração de execução de um processo

É possível definir a duração de execução esperada para um processo. Se o tempo de execução do processo ultrapassar a duração definida, a guia Operações exibirá um indicador vermelho (aviso).

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione uma pasta e, em seguida, selecione um objeto de processo para o qual deseja definir a duração de execução.

**Observação:** se o objeto de processo apropriado não existir, crie-o. Para obter mais informações sobre a criação de um processo, consulte [Criando processos](#) (na página 97).

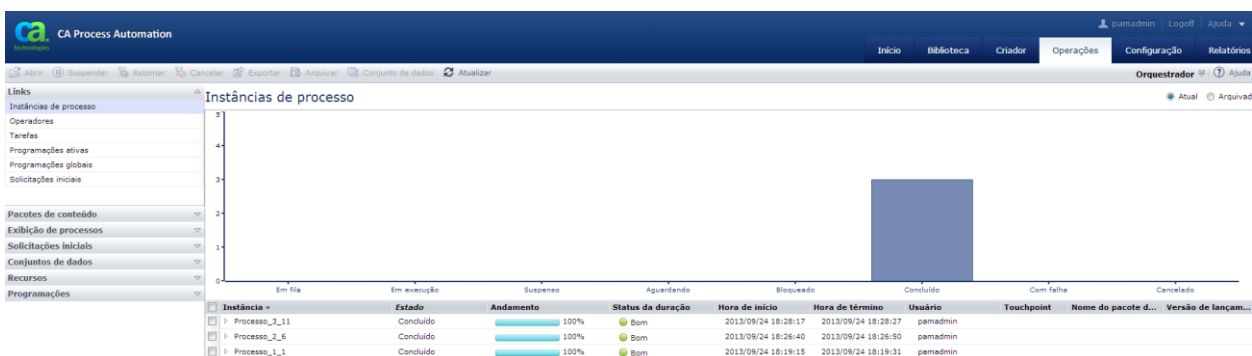
3. Clique com o botão direito do mouse no objeto de processo e, em seguida, selecione Propriedades.
4. No painel Propriedades, clique na guia Duração.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campos e exemplos.

5. Marque a caixa de seleção Ativado.  
O produto ativa os campos da guia Duração.
6. Defina a duração do processo esperada.
7. Defina com quanto tempo de antecedência o produto avisará o usuário se a execução do processo exceder a duração esperada.
8. Clique em Salvar.

O Status da duração exibe um dos seguintes indicadores de status para a instância de processo em tempo de execução associada:

- **Vermelho:** o processo executado excedeu a duração esperada.
- **Amarelo:** o processo está sendo executado, mas está próximo do fim da duração esperada.
- **Verde:** o processo foi executado com a duração esperada. O status será verde quando um processo for concluído e enquanto um processo estiver em execução e não estiver no status amarelo ou vermelho.
- **Nenhum indicador:** o processo não tem uma duração esperada definida.

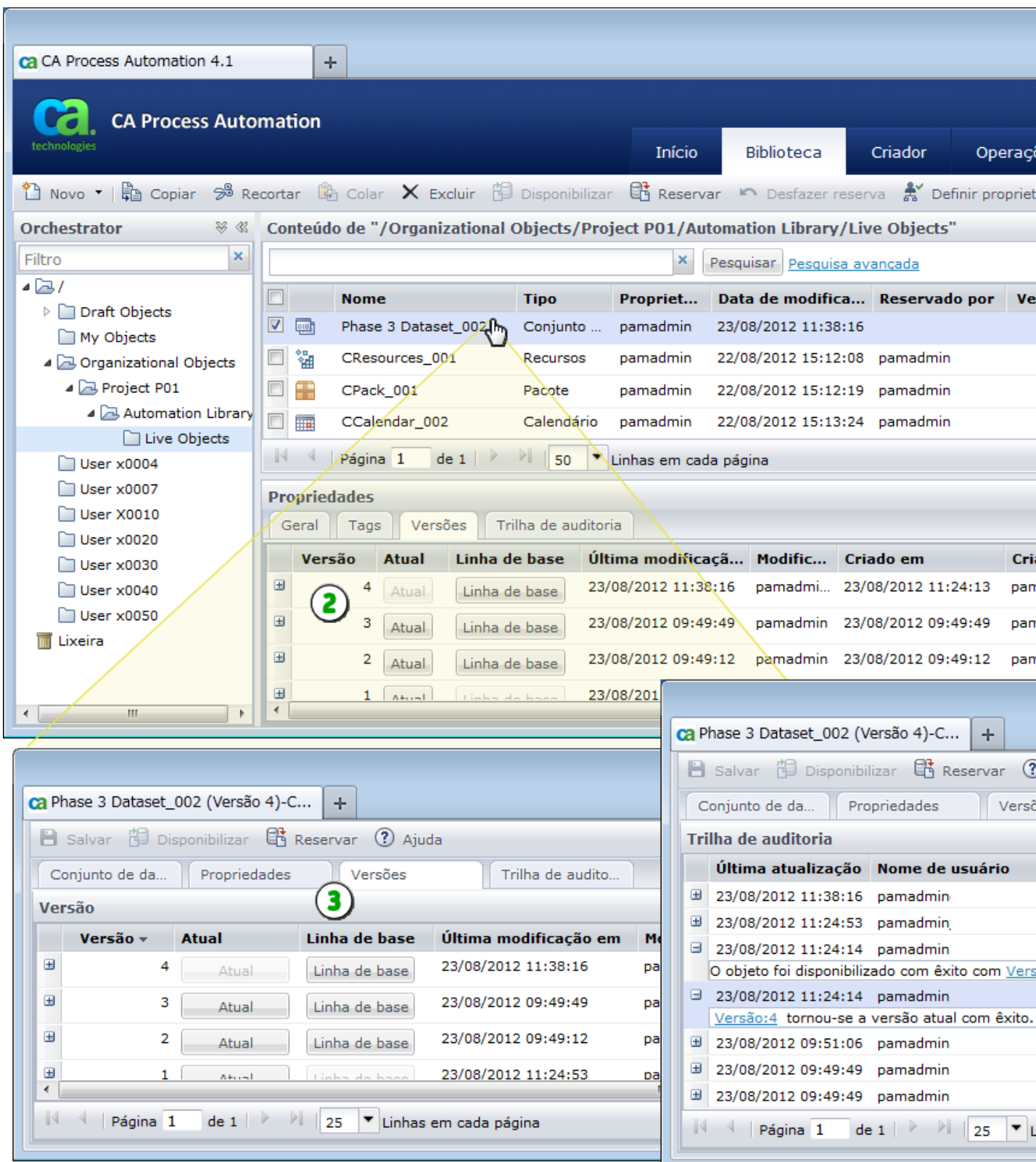


## Versões

O CA Process Automation usa um sistema de controle de versão que rastreia as alterações dos objetos na sua biblioteca. O sistema de rastreamento de versão sempre mantém uma única versão *atual* de cada objeto. Quando você abrir um objeto, o aplicativo usará automaticamente o objeto que você designou como a versão atual. O aplicativo retém outras versões de um objeto para backup ou arquivo. É possível abrir versões arquivadas de um objeto para visualizar, editar ou designar como a versão atual. O proprietário do objeto ou outro usuário que tenha permissão suficiente pode alterar qual versão é a atual.

Você pode reservar e editar qualquer versão de um objeto. Quando as edições estiverem concluídas, será possível disponibilizar as alterações para aquela versão ou para uma nova versão do objeto. Para evitar que os usuários alterem uma versão específica, é possível designá-la como uma versão de *linha de base*. Uma versão de linha de base pode ser usada como qualquer outra versão. Também é possível reservar uma versão base e editá-la, mas só é possível disponibilizar o objeto como uma nova versão. As versões base são um benchmark ou modelo para desenvolvimento adicional.

Para liberar uma nova versão de um processo com todos os seus componentes, defina o atributo de versão da release para o processo e cada objeto associado. A [versão da release](#) (na página 454) geralmente identifica uma versão de um objeto que você implanta em um ambiente de produção. O CA Process Automation mantém o atributo de versão da release que foi definido antes da exportação no ambiente de importação. Você define um atributo de versão da release para cada objeto, adiciona cada objeto à pasta e inicia a exportação da pasta. Se você exportar a pasta como pacote de conteúdo, a versão da release não será modificável no ambiente de produção. O processo de importação automaticamente define a linha de base da versão da release de um pacote de conteúdo para evitar que os usuários disponibilizem alterações na versão importada.



---


Item:	Descrição:
1	<b>Versões do objeto:</b> para cada objeto, o navegador da biblioteca exibe o usuário que reservou o objeto e as versões atuais e mais recentes do objeto. Neste exemplo, a versão atual do conjunto de dados selecionado é 2, que é a versão que o usuário Design_01 reservou. A versão mais recente do objeto é a versão 4.
2	<b>Guia Versões do painel Propriedades:</b> a guia Versões do painel Propriedades lista a versão original e todas as versões subsequentes de um objeto. É possível ver qual versão é a atual, qual versão ou versões são as linhas de base e as informações de criação e modificação. A guia Versões do painel Propriedades também fornece um campo Versão da release editável. Opcionalmente, você pode executar a seguinte ação: <ol style="list-style-type: none"><li data-bbox="613 688 1443 751">1. Adicionar um valor específico à release ao campo Versão da release, como parte da preparação para exportação.</li><li data-bbox="613 772 1443 793">2. Selecionar essa versão do objeto para exportação.</li></ol> <b>Observação:</b> se você tiver exportado o objeto como uma parte de um pacote de conteúdo, o campo Versão da release não será modificável quando o objeto for importado para um novo ambiente. Uma dica de ferramenta sobre o campo indica que ele está bloqueado.
3	<b>Guia Versões em um Editor de objetos:</b> quando você abrir ou editar um objeto, a guia Versões permitirá a seleção de uma única versão atual e de uma ou mais versões base. As outras informações são as mesmas exibidas no painel Propriedades.
4	<b>Guia Trilha de auditoria no Editor de objetos:</b> a guia Trilha de auditoria fornece um histórico de todas as alterações feitas em cada objeto de automação.


---

## Noções básicas sobre versões


Consulte as imagens e os exemplos a seguir para aprender conceitos essenciais sobre como trabalhar com versões.



### Símbolos:

 A versão do objeto atual.

 A versão de trabalho.

Descartar a versão de trabalho.

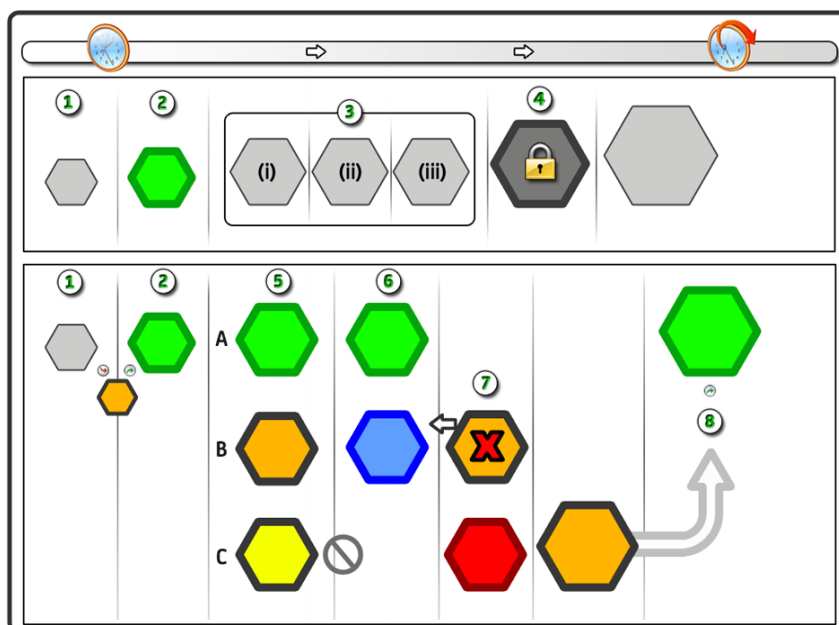
 Abrir uma versão anterior.

 A versão reservada pelo usuário 1.  
 A versão mais recente reservada pelo usuário 2.






 A versão





















A versão de linha de



### Item: Descrição:

- 1 **Sequência típica de controle de versão:** um novo objeto de automação é iniciado na  versão 0. O objeto é reservado como a  versão de trabalho 0, com a opção de aplicar alterações pendentes à mesma versão 0 ou à nova versão 1.
- 2 **Versão atual:** nessa sequência típica, a  versão de trabalho 0 é disponibilizada como nova versão 1. A nova versão também é normalmente designada como a  versão atual. A versão atual é usada quando o processo é realmente executado. A  versão 0 original agora é considerada apenas um backup ou um arquivamento.

**Item: Descrição:**

- 3 **Alterações aplicadas à mesma versão:** neste exemplo, a  versão 2 está reservada várias vezes. Todas as vezes, o usuário optou por aplicar as alterações à mesma versão, em vez de criar uma nova. Embora tenham sido feitas alterações nas  versões de trabalho 3(0), 3(i) e 3(ii), o usuário decidiu agrupar todas elas na mesma versão única. O usuário não poderá voltar para exibir o objeto entre as alterações 3(0) e 3(i). Não há nenhuma versão separada que permita um retorno do usuário.
- 4 **Versão de linha de base:** a versão 4 foi designada pelo usuário como uma  versão de linha de base. Qualquer versão de trabalho reservada deve ser disponibilizada como uma nova versão, a versão 5. Ao especificar uma ou mais versões de linha de base, você bloqueia essencialmente cada versão. Essa prática impede qualquer modificação nas versões reais de linha de base. Elas só podem ser usadas como modelos ou objetos de classe ancestrais para facilitar a criação de novas versões.
- 5 **Versões atuais, de trabalho e abertas:** a  versão atual 5 tem três ramificações simultâneas que os usuários podem seguir. Na ramificação A, as alterações são aplicadas como  uma versão 6 nova e atual. Na ramificação B,  a versão de trabalho 5 pode ser disponibilizada como uma  nova versão 6, com a  versão 5 permanecendo como a versão atual. Na ramificação C, o usuário decide  abrir a versão 5. O usuário pode exibir a versão, mas não pode salvar alterações nela.
- 6 **Vários usuários:** embora os processos em execução utilizem a  versão atual 6, também é possível reservar o objeto para o  usuário 1 com alterações pendentes na  versão de trabalho 7. Embora os processos em execução usem a  versão 6 atual, as alterações mais recentes no objeto podem ser feitas pelo usuário 2 e refletidas como  a versão 7 *mais atual*.
- 7 **Descartar a versão de trabalho:** as alterações feitas na  versão de trabalho 7 podem ser descartadas. De modo semelhante a uma operação Desfazer reserva, essa operação reverte o objeto à última versão disponibilizada.
- 8 **Nova versão atual:** a  versão 7 mais recente pode ser reservada como a  versão de trabalho 7. Durante a disponibilização, o usuário poderá optar por torná-la a  nova versão 8 atual.

**Definir a versão do objeto como Atual ou Versão base**

A guia Versões exibe informações sobre as versões de um objeto. Você pode executar as seguintes tarefas nessa guia:

- Determinar quando e por quem uma versão específica foi criada e atualizada pela última vez.
- Determinar qual é a versão atual de um objeto e, opcionalmente, designar uma versão diferente como atual.
- Determinar quais versões estão definidas como versão base e, opcionalmente, designar uma ou mais versões como versão base.

O produto inicialmente designa um objeto que você criar como Versão (de trabalho) 0. Quando você salvar o objeto e disponibilizá-lo pela primeira vez, o produto irá designá-lo como Versão (atual) 1. A partir desse ponto, o produto designará uma versão como atual e a usará sempre que o processo for executado. O produto inicia a versão atual por padrão quando você abre o objeto na biblioteca. Por padrão, as versões modificadas dos objetos assumem o status Atual ao serem disponibilizados. Opcionalmente, você pode usar o botão Atual para atribuir o status Atual a qualquer versão anterior. Essa opção permite criar uma versão do objeto que o produto não acessa ou abre por padrão. Não é necessário reservar um objeto para definir a versão atual. Somente o proprietário do objeto ou um usuário na função de administrador de conteúdo do ambiente tem permissões para designar uma versão como atual.

Você pode definir a versão de um objeto como versão base para bloquear novas atualizações. Os usuários podem utilizar a versão base como um modelo para criar uma versão mais recente e disponibilizada. Considere definir uma versão base quando você não prevê que fará mais alterações no objeto. Por exemplo, uma prática recomendada é definir a versão base dos objetos antes de empacotá-los para exportação. A versão base definida pode ser a versão atual ou outra versão.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione uma pasta, clique com o botão direito do mouse em um objeto e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia Versões.  
**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campos e exemplos.
4. (Opcional) Clique no botão Atual ativo de uma versão para defini-la como a versão atual e desativar o botão.
5. (Opcional) Clique no botão Versão base ativo de uma ou mais versões para defini-las como versões base e desativar o botão.

**Observação:** não é possível reverter a definição da versão como versão base. Se você atualizar uma versão base, deverá salvá-la como uma nova versão.

## Definir a versão atual de um objeto

A versão atual de um objeto é a versão padrão usada por CA Process Automation no desenvolvimento ou na produção. Não é necessário reservar um objeto para definir a versão atual, mas você deve ser o proprietário do objeto ou um membro da função de Administrador de conteúdo do ambiente.

### Siga estas etapas:

1. Clique na guia Navegador da biblioteca.
2. Clique duas vezes em um objeto.  
O objeto é aberto em seu editor.
3. Clique na guia Versões.
4. Na linha de uma versão específica, clique no botão Atual.

O botão Atual desativado indica que a versão selecionada é a atual. Você pode alterar a versão atual diretamente clicando no botão Atual em qualquer linha.

## Abrir a versão atual ou a versão de trabalho de um objeto

Quando você clica duas vezes em um objeto reservado na biblioteca, a versão de trabalho é aberta. A versão de trabalho exibe as alterações feitas. Se o objeto não tiver sido reservado, a versão atual será aberta no modo de somente leitura. Reserve o objeto para editá-lo.

## Abrir uma versão selecionada de um objeto

É possível abrir uma versão selecionada de qualquer objeto para exibi-lo ou editá-lo, em vez de abrir a versão atual. Por exemplo, você pode abrir uma versão anterior ou abrir uma nova ramificação do objeto com as alterações mais recentes.

### Para abrir uma versão selecionada de um objeto

1. Clique na guia Navegador da biblioteca.
2. No painel de pastas do Navegador da biblioteca, clique em uma pasta.
3. Clique com o botão direito do mouse em um objeto, escolha Versões e, em seguida, Abrir uma versão.
4. Na caixa de diálogo Editar versão, clique em uma versão do objeto e, em seguida, clique em Abrir.

O objeto é aberto em seu editor.

## Reservar um objeto

Para editar e salvar as alterações em um objeto, reserve-o. A reserva de um objeto permite editar versões do objeto e, ao mesmo tempo, impede que outros usuários o alterem ao mesmo tempo. Você poderá abrir e exibir um objeto sem reservá-lo, mas não poderá editá-lo.

Você poderá reservar um ou mais objetos antes de abri-los.

### Siga estas etapas:

1. No Navegador da biblioteca, selecione um ou mais objetos.
2. Clique em Reservar.

Agora, você pode clicar duas vezes no objeto reservado para abrir e editá-lo.

Também pode reservar um único objeto depois de abri-lo.

### Siga estas etapas:

1. No Navegador da biblioteca, clique duas vezes em um objeto.

O objeto é aberto em seu editor.

2. No editor de objetos ou no Criador de processos, clique em Reservar.

Você poderá reservar, editar, disponibilizar e testar as alterações e, em seguida, reservar e continuar editando o objeto.

## Salvar alterações para um objeto reservado

Salvar a versão de trabalho de um objeto impede que você perca as alterações durante o trabalho. Por exemplo, você poderia acidentalmente fechar o editor de objetos. Salvar as alterações para um objeto reservado afeta somente a versão de trabalho do objeto. Uma nova versão do objeto não é criada, e o objeto não é reservado.

### Siga estas etapas:

1. [Reserve um objeto](#) (na página 86).
2. Edite a versão de trabalho.
3. Clique em Salvar.

Depois que as alterações forem salvas na versão de trabalho, será possível fechar o editor. O objeto continua reservado. A versão de trabalho permanece editável e continua a refletir suas alterações, desde que ela permaneça reservada. Quando você abrir o objeto novamente, o aplicativo automaticamente abrirá a versão de trabalho.

## Descartar alterações para um objeto reservado

Depois de reservar, modificar e salvar um objeto, você pode decidir que não deseja continuar a trabalhar com aquela versão específica. Mesmo que tenha salvo a versão de trabalho inúmeras vezes, é possível executar Desfazer reserva para descartar a versão de trabalho salva.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse no objeto na janela principal e selecione Disponibilizar/reservar.
3. Selecione Desfazer reserva.

O aplicativo ignora as alterações salvas para a versão de trabalho e disponibilizará o objeto.

4. Para continuar a trabalhar com o objeto, reserve-o novamente.

## Disponibilizar um objeto

Ao concluir suas alterações, disponibilize o objeto para salvar as alterações na mesma versão ou em uma nova versão do objeto.

Depois de concluir o procedimento a seguir, o criador permanece aberto e é possível continuar a exibir o objeto no modo somente leitura. Reserve o objeto novamente para fazer mais alterações.

### Siga estas etapas:

1. [Reserve um objeto](#) (na página 86).
2. Edite o objeto.
3. [Salve as alterações para um objeto reservado](#) (na página 86).
4. Na barra de ferramentas do editor de objetos ou na barra de ferramentas do Navegador da biblioteca, clique em Disponibilizar.

A caixa de diálogo Disponibilizar é exibida com o nome do objeto no campo Nome do objeto.

5. Para especificar como disponibilizar suas alterações, defina as seguintes configurações:

#### **Atual**

**Marcada:** fazer da versão disponibilizada a versão *atual* do objeto. A versão atual é aquela utilizada pelo CA Process Automation.

**Desmarcada:** manter a versão atual anterior como a versão atual. Desmarque a caixa de seleção se estiver disponibilizando um objeto cuja criação está em andamento.

#### **Linha de base**

**Marcada:** a versão do objeto que você está disponibilizando não poderá ser alterada durante uma reserva futura. As versões de linha de base somente podem servir de base para novas versões. Você pode reservar uma versão base, alterá-la e disponibilizar o objeto como uma nova versão.

**Desmarcada:** a versão do objeto que você está disponibilizando pode ser alterada durante uma reserva futura.

#### **Versão**

Exibe a versão do objeto que você está disponibilizando.

#### **Versão da release**

Define um indicador exclusivo para a versão de um objeto selecionado para exportação como parte de uma release.

Se o campo Versão da release for somente leitura, o objeto foi importado anteriormente com a versão da release no modo não modificável. A dica de ferramenta fornece detalhes.

Digite um valor para a Versão da release somente quando você estiver preparando a versão do objeto para exportação (individualmente, em uma pasta). Digite um valor específico à release para que não haja conflito com qualquer valor de versão da release de um objeto existente no ambiente de importação.

**Observação:** se ocorrer um conflito, a importação falhará. Especificamente, a importação falha nas seguintes condições:

- O objeto foi importado com a opção Importar como uma nova versão e Manter objeto existente.
- A versão do objeto existente tem o mesmo valor da Versão da release inserido no campo Versão da release.

### Comentários

(Opcional) Insira comentários descritivos para salvar com essa versão do objeto.

### Disponibilizar como

Especifica se deve ser criada uma versão do objeto ou se as alterações devem ser aplicadas à mesma versão.

**Nova versão:** cria uma versão do objeto.

**Mesma versão:** substitui a versão atual do objeto pelas alterações feitas desde que ele foi reservado pela última vez.

6. Clique em Disponibilizar.

**Observação:** se você tentar disponibilizar as alterações de um objeto com versão base definida, o CA Process Automation exibirá uma mensagem de erro.

## Noções básicas sobre linhas de base

A finalidade de se definir a linha de base de um objeto é bloquear um versão.

Considere o caso quando estiver editando a versão de um objeto de automação e salvando as alterações. Depois que você concluir todas as alterações planejadas, defina a versão atual do objeto como a versão base, para evitar futuras alterações nessa versão. É possível sair da versão de uma linha de base, modificá-la e salvar as alterações, mas não é possível entrar no objeto como sendo a mesma versão. Você deve disponibilizar as alterações como uma nova versão.

Por exemplo, suponha que você defina como base a versão 0 de um processo e continue desta maneira:

1. Você sai da versão 0.
2. Você edita a versão 0.
3. Você salva a versão 0.
4. Você disponibiliza a versão 0.
  - Se você disponibilizar as alterações como Nova versão, o CA Process Automation criará a versão com êxito.
  - Se você tentar disponibilizar as alterações como Mesma versão, o produto exibirá a seguinte mensagem:

Você saiu da versão de uma linha de base. Para disponibilizar as alterações, é necessário selecionar Nova versão.

Você não pode substituir a versão de uma linha de base, disponibilizando as alterações para a mesma versão.

Considere definir uma versão base quando você não prevê que fará mais alterações no objeto. Por exemplo, é uma prática recomendada definir a versão base de objetos antes de exportá-los para um pacote de conteúdo. É possível igualar o processo de empacotamento a um processo de liberação. Portanto, antes de liberar um objeto, você pode bloqueá-lo selecionando a versão base na disponibilização final. Você ainda poderá modificar o objeto posteriormente, desde que disponibilize suas alterações em uma versão diferente.

**Observação:** durante a exportação de uma pasta como pacote de conteúdo, o CA Process Automation exporta os atributos de Versão da release no modo não modificável. O processo de importação define a versão base de todos os objetos, de modo que os usuários no ambiente de destino não possam modificar as versões do objeto exportado.

## Criar uma versão da linha de base de um objeto

Para evitar que os usuários alterem uma versão específica, você pode designá-la como uma versão de *linha de base*. Os usuários podem reservar uma versão de linha de base e editá-la, mas só poderão disponibilizá-la como uma nova versão.

Não é necessário reservar objetos para defini-los como linhas de base, mas você deve ser o proprietário ou ter permissão administrativa para o objeto.

### Siga estas etapas:

1. Clique na guia Navegador da biblioteca.
2. Clique duas vezes em um objeto.  
O objeto é aberto em seu editor.
3. Clique na guia Versões.
4. Na linha de uma versão específica, clique no botão Linha de base.

A versão selecionada agora é uma linha de base indicada pela desativação do botão Linha de base. Você pode selecionar várias versões de linha de base de um objeto.

**Observação:** não é possível desfazer ou reverter o status de linha de base de uma versão. Se quiser redefinir uma versão para que ela não seja mais uma linha de base, defina a versão como atual, reserve-a e, em seguida, disponibilize-a como uma nova versão com a caixa de seleção Linha de base desmarcada.

## Excluir ou restaurar um objeto ou uma pasta

Você pode excluir um objeto de automação, uma pasta vazia ou uma pasta que contém objetos quando eles não forem mais necessários. O aplicativo move os itens excluídos para a lixeira.

Se você excluir acidentalmente um objeto necessário (ou uma pasta que contém um objeto necessário), é possível restaurar esse objeto com sua estrutura de pastas.

**Observação:** apenas um administrador pode limpar permanentemente os objetos (e suas pastas) da Lixeira.

### Siga estas etapas:



1. Clique na guia Biblioteca.
2. Selecione um ou mais objetos ou pastas.
3. Clique em Excluir.  
  
Para impedir a perda imediata de dados, os itens serão movidos automaticamente para a Lixeira.
4. No painel de pastas do Navegador da biblioteca, clique na Lixeira.
5. (Opcional) Classifique por Data, Nome ou Tipo, ou use os recursos de pesquisa para localizar itens específicos.
6. Para restaurar objetos ou pastas, selecione-os e clique em Restaurar o item selecionado.
7. Para excluir objetos ou pastas permanentemente, selecione-os e clique em Limpar os itens selecionados.

**Importante:** após limpar objetos ou pastas, não será possível recuperá-los. Ao selecionar pastas ou um conjunto de objetos para eliminação, o processo de eliminação listará todos os objetos reservados. O administrador verificará a lista. Em seguida, o administrador prosseguirá ou cancelará a eliminação, de acordo com o resultado da análise.

## Copiar e colar um objeto ou uma pasta

É possível copiar objetos ou pastas. Você pode usar a versão atual de um objeto como base para objetos configurados de forma semelhante no mesmo orquestrador. Copie a versão atual de um objeto para a mesma pasta ou outra pasta, onde seja possível renomeá-la e editá-la como um novo objeto. Por exemplo, você pode configurar um objeto operador personalizado e adicionar cópias dele em uma biblioteca.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Identifique os objetos a serem copiados na janela principal. Verifique se a versão desejada está marcada como a versão atual.
3. Selecione um ou mais pastas ou objetos de origem ou pastas e clique em  Copiar.
4. No painel da pasta do Navegador da biblioteca, clique em uma pasta de destino na mesma biblioteca de automação e clique em  Colar.

Observe os seguintes resultados:

- O CA Process Automation adiciona os objetos copiados com os mesmos nomes dos objetos originais.
  - Cada objeto possui uma versão, aquela que foi copiada.
  - Se a versão copiada tiver um valor de Versão da release, a cópia manterá esse valor.
  - Se a versão copiada foi definida como versão base, a cópia será definida como versão base.
  - Se esse local for o mesmo da pasta de origem, o CA Process Automation anexará um número a cada nome de objeto (por exemplo, Process\_1).
5. (Opcional) Renomeie os novos objetos.

## Recortar e colar um objeto ou uma pasta

É possível recortar objetos ou pastas e, em seguida, colá-los na mesma pasta ou em uma pasta diferente. Por exemplo, você pode recortar um objeto operador personalizado e colá-lo na mesma biblioteca.

### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Identifique os objetos a serem recortados na janela principal.
3. Selecione uma ou mais pastas ou objetos de origem e clique em Recortar.
4. No painel de pastas do Navegador da biblioteca, clique na pasta de destino localizada na mesma biblioteca de automação e clique em Colar.

Se os nomes dos objetos a serem colados existirem no local de destino, selecione objetos no local de destino para serem substituídos pelos objetos que você recortar.

## Como trabalhar com conteúdo não modificável

O CA Technologies pode liberar conteúdo não modificável. O ícone de negação indica que o conteúdo é não modificável:



O uso de objetos não modificáveis é diferente do uso de objetos modificáveis nos seguintes aspectos:

- Não é possível modificar a versão importada.
- Não é possível copiar e colar a versão importada em outra pasta e, em seguida, modificar a cópia.
- Não é possível modificar a versão de release de um objeto importado.
- Não é possível modificar a versão de release de uma cópia do objeto importado.

### Detalhes de nível de ação no uso de conteúdo não modificável

As ações a seguir são permitidas em conteúdos não modificáveis:

- Copiar o objeto com um novo nome e colá-lo em uma pasta diferente.
- Recortar o objeto e colá-lo em uma pasta diferente.
- Excluir o objeto importado.
- Exibir as propriedades do objeto.
- Executar todas as ações exibidas na lista suspensa Mais ações de objetos importados.

As ações a seguir não são permitidas em conteúdos não modificáveis:

- Edite o objeto.
- Renomear o objeto.
- Disponibilizar ou reservar o objeto.
- Abrir a versão do objeto para edição.
- Editar as propriedades do objeto.
- Definir o proprietário do objeto.
- Exportar o objeto.

Se o conteúdo não modificável for exportado para uma pasta, as seguintes ações não serão permitidas na pasta:

- Importar
- Criar objeto
- Colar



# Capítulo 4: Criando processos

---

Os objetos de processo do CA Process Automation representam graficamente os operadores, as portas, os links, a lógica e as restrições. Cada processo contém uma ou mais cadeias de operadores que podem ser executadas em sequência ou em paralelo. As rotas separam processos pai de qualquer processo secundário disparado. Por exemplo, uma ramificação de um processo que alterna para um subprocesso é geralmente representado em uma rota separada. Os manipuladores de exceção controlam os operadores cancelados ou encerrados devido a erros do sistema ou a condições de saída não identificadas.

Cada processo define a configuração e o gerenciamento de operadores em touchpoints de um ambiente. Os objetos de processo são armazenados com outros objetos em uma biblioteca associada a um orquestrador. Você pode criar objetos de processo no Navegador da biblioteca ou diretamente no Criador. Você também pode abri-los a partir do Navegador da biblioteca. No entanto, os processos são exibidos e editados no Criador de processos.

Esta seção contém os seguintes tópicos:

[Criador de processos](#) (na página 98)

[Operadores e links: os blocos de criação](#) (na página 100)

[Criar um objeto de processo](#) (na página 101)

[Criar um processo](#) (na página 102)

[Operadores de processo](#) (na página 104)

[Portas e links do operador de processo](#) (na página 110)

[Loops e iterações de processo](#) (na página 115)

[Controle de processo](#) (na página 126)

[Rotas de processo](#) (na página 134)

[Versões de processos](#) (na página 138)

[Documentar um processo](#) (na página 138)

[Conteúdo Self-Contained](#) (na página 140)

[Navegar até uma parte específica de um processo](#) (na página 143)

[Multilocação e CA Process Automation](#) (na página 144)

[O editor de código do CA Process Automation](#) (na página 146)

## Criador de processos

O Criador de processos fornece um ambiente de desenvolvimento integrado que lhe permite arrastar e soltar operadores e links para criar processos. O Criador de processos também fornece recursos de configuração, teste e depuração de propriedades e conjuntos de dados.

Use o Criador de processos para:

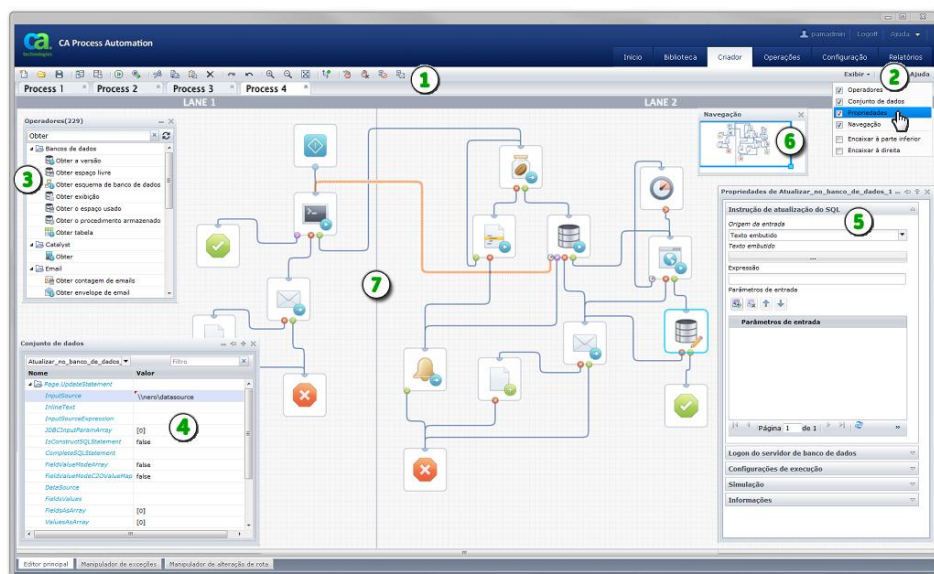
- Criar e editar processos.
- Executar, monitorar e controlar a execução dos processos nos ambientes de produção ou de teste.
- Modificar uma instância de um processo em execução para recuperar a partir de um incidente.

Adicionar um operador, arrastando-o da paleta Operadores para o layout de design. Você também pode exibir e configurar diretamente as propriedades ou os valores de conjunto de dados para um operador ou um processo.

O layout do Criador de processos inclui os seguintes elementos de processo:

- **Tela:** arraste um operador para exibir guias a fim de ajudar seu operador a se ajustar às posições na grade.
- **Rotas:** o processo de exemplo mostra duas rotas para a organização de segmentos. Você pode adicionar mais rotas, mesclá-las ou remover rotas. Você também pode consultar as rotas como *diagramas de rota*.
- **Operadores:** as entidades funcionais dentro de um processo. Cada operador, exceto os do final, mostrará suas portas de saída e outros ícones de status menores.
- **Portas:** esses pequenos pontos de conexão representam as portas de saída para cada operador. Cada operador, exceto o operador Iniciar, também tem uma única porta de entrada.
- **Links:** essas linhas conectam a porta de saída de um operador à porta de entrada de outro operador. Você pode personalizar a aparência dessas linhas.
- **Editores de manipuladores:** além do Editor principal, o criador também inclui duas outras guias ao longo da parte inferior para a edição de manipuladores de exceção e de alteração de rota.

O gráfico a seguir ilustra o Criador de processos.




---

**Item: Descrição:**


---

- 1** **Barra de ferramentas do Criador e guias de processos:** quando você abre um processo a partir do Navegador da biblioteca, a guia Criador é exibida. Cada processo aberto é exibido em sua própria guia. Use os botões da barra de ferramentas para reservar, editar, aplicar mais ou menos zoom, salvar, testar e disponibilizar o objeto de processo selecionado. Você pode copiar e colar os operadores de uma guia de processo em outra. A barra de ferramentas também apresenta ícones para a criação de um processo ou a abertura de um processo existente (as imagens da tela podem variar).
  - 2** **Menu Exibir:** use as configurações de Exibir na parte superior direita para mostrar ou ocultar as paletas Operadores, Conjunto de dados, Propriedades e Navegação. Você também pode acomodar as paletas de propriedades e conjuntos de dados à direita ou na parte inferior. Clique em Destacar para abrir a página atual em sua própria janela no navegador a fim de maximizar a sua exibição.
  - 3** **Paleta Operadores:** arraste e solte os operadores dessa paleta até o layout do seu processo. Você também pode inserir critérios de pesquisa (por exemplo, "Get") para filtrar os operadores não correspondentes.
  - 4** **Paleta Conjunto de dados:** use essa paleta para exibir, editar e adicionar variáveis no processo ou em conjuntos de dados do operador.
  - 5** **Paleta Propriedades:** use essa paleta e seus botões e janelas adicionais para gerenciar as propriedades do operador selecionado no momento.
  - 6** **Paleta Navegação:** use essa paleta para navegar até regiões específicas dentro de processos grandes com várias rotas. Para sua conveniência, você pode aplicar uma panorâmica em qualquer direção da paleta em vez de rolar o layout principal do criador para cima ou para baixo.
  - 7** **Criador de processos:** o design do processo real é exibido nessa área de trabalho, tela ou layout. O Criador de processos inclui a grade de plano de fundo e uma ou mais rotas.
-

## Operadores e links: os blocos de criação

A estrutura geral de um processo do CA Process Automation consiste em dois itens básicos: operadores e links. Os *operadores* executam tarefas ou testes condicionais. Os *links* conectam os operadores e determinam a sequência de processamento.

A lista a seguir descreve alguns exemplos de operadores:

- Os operadores Gerenciamento de arquivos monitoram tamanhos de arquivo, padrões e outros parâmetros relativos a arquivos em um sistema.
- Os operadores Transferência de arquivos usam FTP para executar transferências de arquivos e operadores de arquivos remotos.
- Os operadores Email notificam os administradores do sistema em caso de erros ou outros estados condicionais que exigem intervenção humana.
- Os operadores de interface Utilitários de rede obtêm, atualizam e monitoram variáveis de SNMP, além de enviar SNMP traps para dispositivos de rede e monitores.
- Os operadores de recurso representam os recursos em fornecimento limitado.
- Os Serviços web (métodos SOAP) representam uma interface entre os produtos de terceiros e o CA Process Automation.

Os links conectam os operadores e executam o fluxo de processamento. O ponto de interseção de um operador e um link é chamado de *porta*. Um link é originado na porta de saída de um operador e termina na porta de entrada de outro operador.

Quando um processo entre dois operadores é executado, a sequência de processamento pode ser resumida da seguinte forma:

- Ativar o primeiro operador.
- Aplicar lógica, obter um resultado e ir até a porta de saída apropriada.
- Ativar o segundo operador.

Um operador pode ter várias portas de saída para processar vários resultados. Cada link de saída pode iniciar uma ramificação separada de operadores em um subprocesso ou em um processo *filho*. Portas de entrada e saída aplicam uma sequência linear na qual os operadores são processados. Elas podem direcionar o fluxo de processamento para uma determinada ramificação de um processo, dependendo do resultado de um único operador.

Para direcionar o processamento com base nos resultados de vários operadores, use os seguintes operadores:

- O operador *Loop*.
- O operador *Exceção* no modo Manipulador de exceções.
- Operadores condicionais, como o operador *E* ou o operador *Ou*.


Esses operadores permitem criar várias ramificações de entrada e saída para outros operadores. Também é possível criar links que dependem dos resultados de vários operadores.

## Criar um objeto de processo

Use o Navegador da biblioteca para criar um objeto *Processo* em uma pasta. Você também pode criar um processo no Criador e especificar uma pasta quando salvá-lo. Crie um objeto *Processo* para cada sequência de automação, fluxo ou subprocesso distinto que você deseja automatizar. Após a criação do objeto *Processo*, crie o processo adicionando operadores e conectando-os aos links.

**Observação:** para definir direitos e executar outras tarefas de objeto, a prática recomendada é organizar os objetos em pastas. Não crie objetos no nível raiz, pois não é possível gerenciá-los como um grupo.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. No painel de pastas do Navegador da biblioteca, selecione uma pasta.
3. Na barra de ferramentas, clique em Novo e, em seguida, escolha Tipo .
- Um novo objeto de processo é exibido com o nome padrão. O processo é automaticamente reservado para você capturar suas alterações exclusivas.
4. Clique no nome do processo e altere-o para um nome exclusivo. O nome pode ser diretamente editado até que você cancele a seleção. Para editá-lo novamente, clique com o botão direito do mouse e escolha Renomear.

Você criou um novo objeto de processo. Em seguida, você pode criar o processo.

## Criar um processo

Execute as etapas descritas em [Criar um objeto de processo](#) (na página 101) primeiro. Depois de entender os conceitos e as etapas de criação dos primeiros novos processos, você poderá editar qualquer processo facilmente. Use o Criador de processos para criar e configurar todos os objetos de processo.

### Siga estas etapas:

1. Clique duas vezes no processo na Biblioteca ou abra-o a partir do Criador de processos.  
O Criador de processos será aberto com um conjunto padrão de operadores básicos de início e interrupção.
2. Use o Criador de processos para criar e configurar o processo. Isso inclui a adição de operadores, portas e links. Consulte os tópicos remanescentes nessa seção para orientá-lo.
3. Para configurar as propriedades de um processo, clique em qualquer espaço livre na tela e escolha Propriedades no menu Exibir da barra de ferramentas.

A paleta Propriedades é exibida. As propriedades do processo determinam o comportamento padrão de todos os operadores adicionados ao processo.

### Configuração

Especifica se os diagramas de raia devem ser exibidos na horizontal ou vertical.

### Vincular

Especifica peso, cor e forma das linhas que vinculam os operadores.

### Simulação

Especifica as opções de simulação padrão para os operadores adicionados a um processo. Você também pode substituir essas configurações para um operador específico.

### Opções de rótulo

Especifica as opções de exibição padrão para os operadores adicionados a um processo. Você também pode substituir essas configurações para um operador específico.

4. Ao terminar, clique em Aplicar para exibir as alterações.
5. Continue com todas as outras tarefas nessa seção deste guia.
6. Quando concluir a edição de um objeto, clique em Salvar e, em seguida, em Disponibilizar.

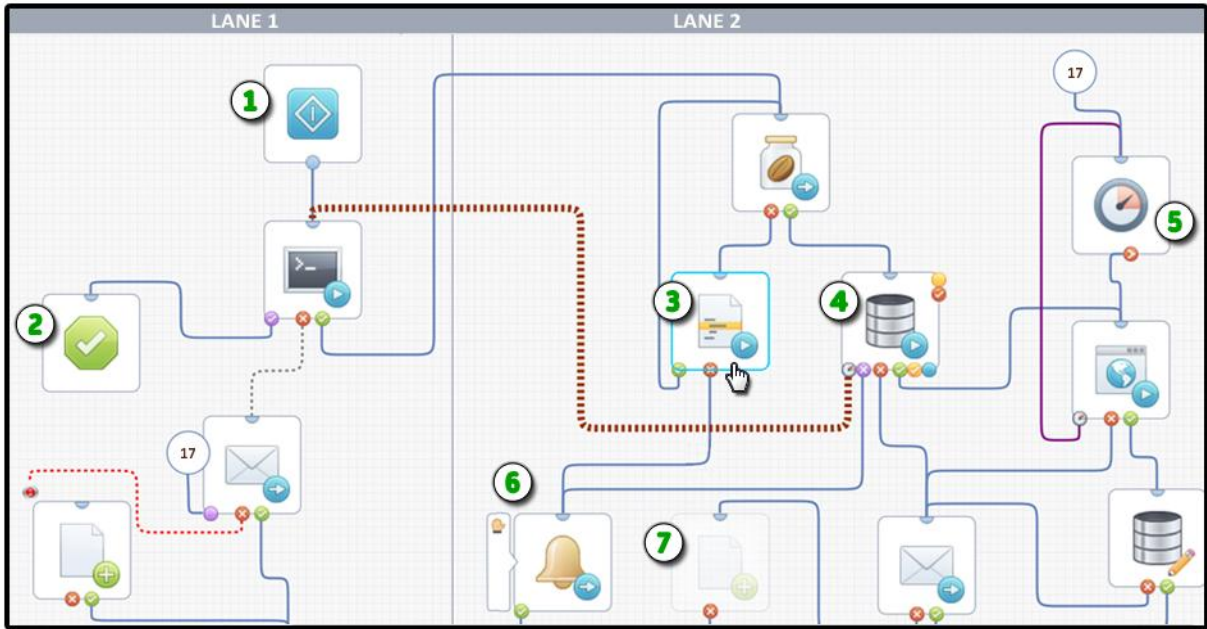
### Dicas de criação de processos

Lembre-se das dicas a seguir ao trabalhar com processos:

- É possível repetir qualquer um dos tópicos em todo este capítulo em praticamente qualquer ordem. Por exemplo, adicionar um diagrama de rota, adicionar uma porta ou conectar-se a um link.
- Como regra básica, você pode editar um processo usando as mesmas etapas de edição de qualquer objeto de automação na biblioteca. Consulte [Trabalhando com objetos](#) (na página 64).
- Sempre reserve o processo para impedir que outros usuários substituam suas alterações. Você pode reservar um processo antes de abri-lo (no Navegador da biblioteca) ou depois de abri-lo (no Criador de processos).
- O CA Process Automation sempre executa a versão atual de um processo; no entanto, se você tiver reservado um objeto, o CA Process Automation será inteligente o bastante para executar sua versão de trabalho.
- Ao disponibilizar um processo alterado, decida como suas versões serão manipuladas. Você pode substituir a versão aberta ou criar uma nova versão distinta separada.
- Antes de testar as alterações em um processo, disponibilize a versão editada que você reservou anteriormente.

## Operadores de processo

Os operadores de processo aplicam funções ou executam ações para produzir um resultado. O conjunto de resultados de uma série de operadores determina o fluxo automatizado por todo o processo.



**Item:**    **Descrição:**

**Operador Iniciar:** todos os processos incluem pelo menos um operador *Iniciar* sem nenhuma porta de entrada e com uma única porta de saída. Embora seja raro, é possível ter mais de um operador Iniciar em um processo.

**2** **Operador Interrupção bem-sucedida:** todos os processos incluem pelo menos um operador *Interromper*. Os operadores Interromper são fornecidos em dois resultados ou tipos: *êxito* (mostrado aqui) ou *falha*.

**3** **Operador selecionado:** a borda do operador atualmente selecionado é exibida como uma linha tracejada. A paleta Propriedades exibe as configurações do operador selecionado no momento.

**4** **Operador de consulta de banco de dados com várias portas de saída:** à medida que você adiciona portas, elas são exibidas ao redor das bordas inferior e direita do operador. Cada porta representa o fluxo de processo para um determinado resultado.

**Observação:** clique com o botão direito do mouse em um operador para adicionar uma porta.

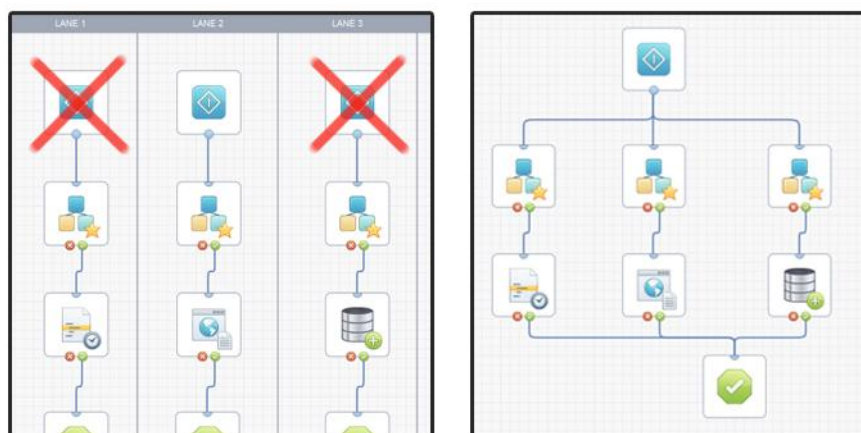
**5** **Operador Atrasar com porta Após:** os operadores Atrasar não incluem portas *Concluído* ou *Tempo limite* bem-sucedidas. Além de uma porta de resultado personalizada, eles apresentam apenas duas portas padrão: *Após* e *Falha*. Operadores diferentes incluem portas distintas.

Item:	Descrição:
6	<b>Operador Enviar evento com indicador de ponto de interrupção:</b> os indicadores de status aparecem quando você clica em Definir ponto de interrupção.
7	<b>Operador Desativado:</b> quando você desativa um operador, o criador atribui a ele uma aparência transparente. Você pode decidir ativá-lo posteriormente.

## Operador Iniciar

Um operador *Iniciar* padrão será automaticamente aplicado quando você criar e abrir um novo processo. O operador Iniciar não possui porta de entrada. Configure as propriedades do operador Iniciar na paleta Propriedades. Por exemplo, você pode alterar o nome, especificar um ícone personalizado ou alterar o local e o texto no rótulo do ícone.

O operador Iniciar inicia uma cadeia de operadores em qualquer processo. Quando um novo processo é iniciado, ele imediatamente ativa o operador Iniciar. Todos os operadores devem ser vinculados direta ou indiretamente a essa cadeia de operadores. Um processo pode ter vários operadores Iniciar para iniciar vários caminhos de processamento. Esses caminhos também são conhecidos como *ramificações* ou *subprocessos*. No entanto, como regra geral, um processo realmente precisa apenas de um operador Iniciar. Conforme ilustrado no exemplo à direita, um design mais eficiente usa o mesmo operador Iniciar e, talvez, até o mesmo operador Interrupção bem-sucedida. Vários links podem compartilhar o mesmo operador.



**Observação:** executar várias ramificações em paralelo com vários operadores Iniciar pode resultar em confusão das entradas de log. Os logs podem ser difíceis de serem analisados, pois a sequência de execução de ramificações com vários inícios pode ser exibida em ordem aleatória. Na maioria dos casos, use um único operador Iniciar para cada processo.

## Adicione os operadores a um processo

Depois de colocar o operador Iniciar em um processo, arraste o próximo operador e associe os links que determinam o fluxo do processo.

### Siga estas etapas:

1. Execute um dos procedimentos a seguir para abrir um processo
  - a. No Navegador da biblioteca, clique duas vezes em um objeto Processo.
  - b. Na barra de ferramentas do Criador, clique em Abrir.

A guia Criador de processos é exibida. Se ainda não estiver reservado, clique em Reservar.
2. Arraste um operador da paleta Operadores para o layout do processo, em qualquer local abaixo ou à direita do operador *Iniciar*. Os operadores são organizados em pastas para cada *módulo* ou categoria de operador com suporte no CA Process Automation. Digite o nome de um operador no campo Pesquisar para procurar operadores correspondentes.
3. Repita a etapa anterior quantas vezes for necessário para criar uma cadeia de operadores.
4. Crie o primeiro link na cadeia. Clique na pequena porta de saída sob o operador *Iniciar* e arraste o link para a porta de entrada do operador de destino.
5. Continue a conectar logicamente as portas de saída para as portas de entrada dos operadores em sequência.
6. Configure cada operador, clicando nele duas vezes para abrir sua paleta Propriedade.

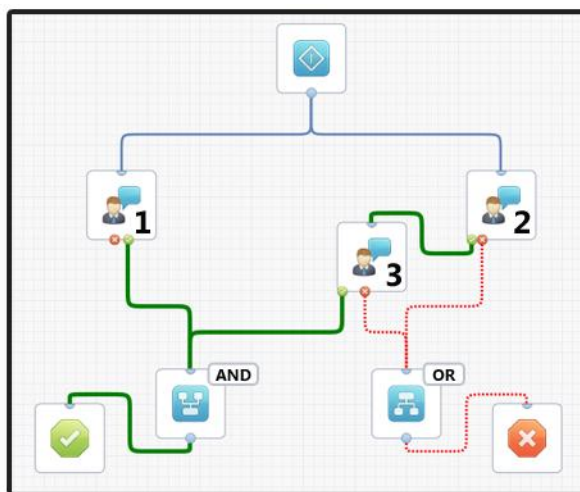
## Operadores lógicos

Os operadores lógicos (*E*, *Ou*) são usados para sincronizar, unir e dividir o processamento de acordo com as condições. Nessa versão, os operadores E e Ou estão disponíveis na pasta Padrão, na paleta Operadores.

**Observação:** a partir do CA Process Automation r4.0, o operador Derivação não está mais disponível. Quando importado ou migrado de uma versão anterior para esta, o operador Derivação é convertido em um operador Ou.

### Exemplo: lógica condicional

O exemplo de processo a seguir mostra oito operadores e nove links (dois azuis, quatro verdes e três vermelho).



- O operador Iniciar na parte superior inicia o processamento simultâneo dos dois primeiros operadores vinculados (Tarefa 1 e Tarefa 2).
- A Tarefa 3 é iniciada *somente* depois que a Tarefa 2 é concluída com êxito.
- O operador E nos links de saída da Tarefa 1 e da Tarefa 3 será ativado somente depois que *ambas* Tarefas, 1 e 3, forem concluídas com êxito. Esse caminho termina no operador verde, Interrupção bem-sucedida.
- O operador nos links de saída da Tarefa 2 e da Tarefa 3 será ativado após uma saída anormal da Tarefa 2 *ou* da Tarefa 3. Esse caminho termina no operador vermelho, Interrupção com falha.
- Os operadores Controle de processo em execução em um orquestrador executam um processo. É possível executar tarefas de processo individuais em qualquer touchpoint do agente que esteja no mesmo ambiente do orquestrador. Por exemplo, a Tarefa 1 podem fluir para um touchpoint do Windows enquanto a Tarefa 2 e a Tarefa 3 podem resultar em um touchpoint UNIX.

## O operador E

O operador *E* define um ponto de sincronização entre todos os links de entrada vinculados a ele. Os links de saída de um operador *E* serão ativados somente após a ativação de todos os links de entrada vinculados a ele. Use um operador *E* para sincronizar várias ramificações de um processo quando todas as ramificações tiverem que ser concluídas antes do início de uma ou mais ramificações adicionais.

## O operador OU

O operador *Ou* ativa os links de saída quando qualquer um dos links de entrada é ativado. O operador *Ou* também pode ser implementado com um único link de entrada para ativar dois ou mais links de saída para ramificações paralelas separadas.

Não é preciso usar um operador *Ou* para implementar uma condição lógica "ou" em um processo. Dois ou mais links, inserindo o mesmo operador, funcionam da mesma forma que um operador *Ou*. No entanto, para diminuir a confusão, bem como para documentar a lógica em um processo, é recomendável utilizar um operador *Ou* em vez de mesclar links em uma operação. A propriedade *Nome* em um operador *Ou* permite que você rotule o operador e o documento no contexto da sequência lógica de operadores no processo.

## Operador Interromper: êxito ou falha

Encerre um processo, vinculando o operador final de uma sequência a um operador Interromper. Um processo pode ter vários operadores Interromper em ramificações diferentes. Um operador Interromper processado em qualquer ramificação não possui saída e encerra o processamento de todo o processo. Os operadores Interromper podem ser opcionalmente configurados para encerrar um loop de chamada em outro processo.

### Siga estas etapas:

1. Abra e reserve um processo.
2. Na paleta Operadores, arraste um operador *Interrupção bem-sucedida* ou *Interrupção com falha* até o seu processo.
3. Clique duas vezes no operador Interromper para abrir a paleta Propriedades.
4. Na seção Parar, especifique o valor inteiro retornado pelo processo no campo Resultado e escolha um valor para o campo Tipo final.
  - Para finalizar o processo normalmente, clique em *Interrupção bem-sucedida* e digite o valor 1 para Resultado. O valor 1 para Resultado causará a saída de um processo concluído quando o processo for chamado de um operador Iniciar processo em um processo pai.
  - Para finalizar o processo de forma anormal, clique em *Interrupção com falha* e digite o valor 0 para Resultado. O valor 0 para Resultado causará a saída de um processo cancelado quando o processo for chamado de um operador Iniciar processo em um processo pai.
5. (Opcional) Em vez de usar estas configurações padrão para o resultado, é possível especificar qualquer expressão que retorne um número inteiro. A expressão deve retornar um número inteiro diferente de zero, para indicar que o processo foi concluído normalmente, ou zero, para indicar que o processo foi concluído de modo anormal.
6. Marque a caixa de seleção *Interromper loop de chamada* para interromper um loop de chamada. Se o processo tiver sido chamado por meio de um operador em loop em outro processo, essa opção interromperá o loop quando o processamento retornar dessa interrupção para o outro processo. Desmarcar essa caixa de seleção permite a continuidade de um loop de chamada.
7. Marque a caixa de seleção *Ignorar tarefas em execução (interrupção imediata)* para interromper o processamento do processo imediatamente quando a parada for executada. Isso interromperá todos os outros operadores que ainda estejam ativos em outro local no processo. Desmarque essa caixa de seleção para permitir que os operadores em andamento sejam encerrados normalmente antes da interrupção do processo.

## Portas e links do operador de processo

Os links entre os operadores definem dependências. Os links atuam nos resultados produzidos por cada operador. Os links definem a ordem e a lógica de um processo durante seu fluxo.

Tipos diferentes de ações têm diferentes resultados ou efeitos predefinidos:

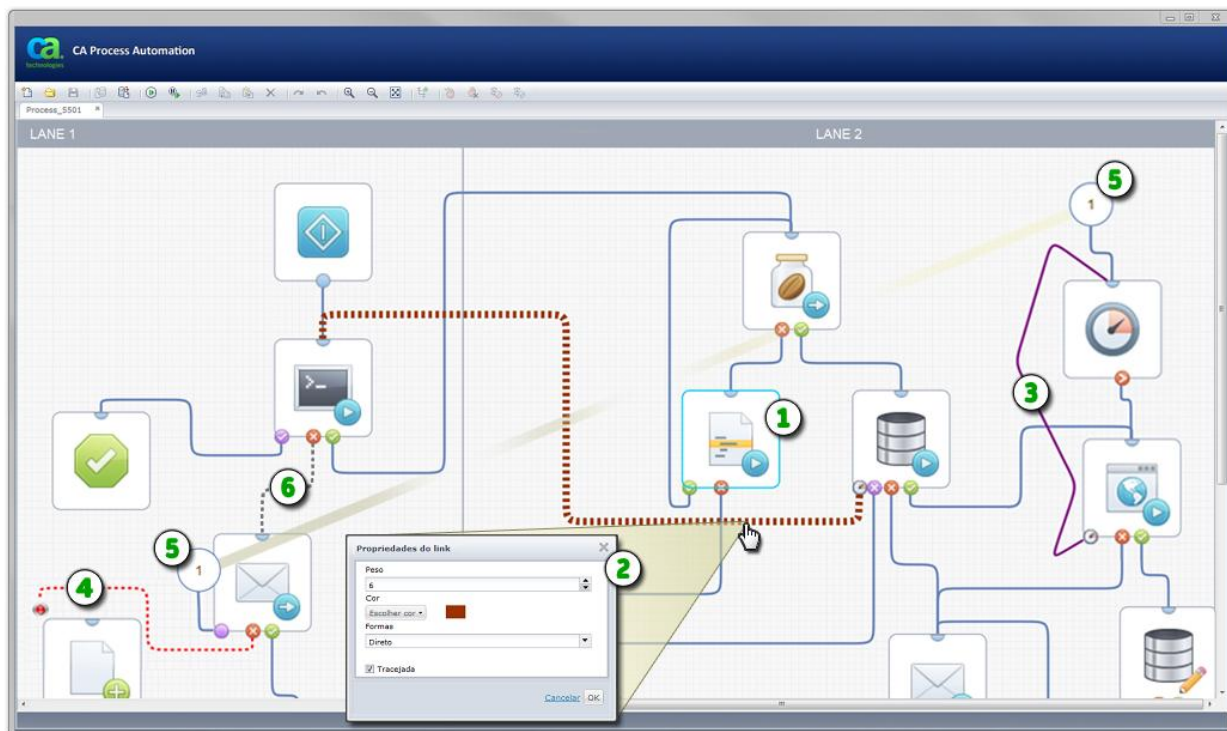
- Com êxito.
- concluído
- Cancelado
- Com falha
- Tempo limite

O aplicativo calcula esses resultados para determinar as próximas condições de saída, portas e links a serem ativados, em uma sequência lógica. Por exemplo, você pode adicionar uma porta personalizada em alguns operadores e pode definir a porta a ser ativada quando uma expressão retornar um valor Verdadeiro.

As condições de saída em um operador não são mutuamente exclusivas. Se o produto avaliar mais de uma condição de saída como Verdadeira, todas as condições de saída serão processadas. O processamento de várias condições de saída em um único operador pode iniciar o processamento simultâneo subsequente de várias ramificações.

Quando um processo é executado, o produto ativa seus operadores apenas uma vez. Quando um link leva a um operador ativado anteriormente, o produto não reprocessa o operador de destino e a ramificação que o link estende termina.

**Observação:** os links são unidos aos operadores em pequenos pontos de conexão chamados *portas*.




---

**Item: Descrição:**

- 1** **Operador selecionado:** clique em um operador para exibir suas variáveis do conjunto de dados, páginas e propriedades. Clique com o botão direito do mouse em um operador para adicionar uma porta de saída.

---

  - 2** **Propriedades do link:** proceda de uma das seguintes maneiras para ajustar a aparência de um link:

    - Clique duas vezes em um link.
    - Clique com o botão direito do mouse em um link e selecione Propriedades do link

Selecione a espessura, a cor, a forma e a aparência tracejada de cada link.

---

  - 3** **Formato de linha do link:** em vez de linhas ortogonais ordinárias, esse link roxo será exibido com segmentos em linha reta. Você pode alongar e posicionar todos os links conforme necessário.

---

  - 4** **Link parado:** esse link foi forçado a parar. Como exemplo, considere um processo que está em loop, aguardando por um evento, processando esse evento e em loop novamente várias vezes. Quando uma ramificação paralela do processo determina que o processo de loop original deve parar, ela pode utilizar a porta e o link parados para interromper o loop.

---

  - 5** **Link interrompido:** interrompa um link para dividir uma longa rota sinuosa em dois stubs numerados. Os links divididos são mais fáceis de visualizar e gerenciar. Para associar novamente os stubs numerados, clique com o botão direito do mouse no número do link circulado e selecione Associar link.
-

---

Item:	Descrição:
-------	------------

---

6

**Link desativado:** este link cinza tracejado indica que ele está temporariamente desativado. Clique com o botão direito do mouse no link para reativá-lo.

---

## Adicionar portas e links do operador

Coloque um link entre os operadores para estabelecer o fluxo lógico. Por exemplo, vincule o operador Iniciar ao próximo operador para iniciar um fluxo de processo. Os links conectam uma das portas de saída em um operador à única porta de entrada fixa de outro.

### Siga estas etapas:

1. Abra um processo.

A guia Criador de processos é exibida. Se ainda não estiver reservado, clique em Reservar.

2. Identifique um operador de origem, sua porta de saída e a porta de entrada de um operador de destino.

3. No operador de origem, se a porta de saída desejada não aparecer, clique com o botão direito do mouse no operador e, em seguida, clique na porta que deseja adicionar. Os tipos de porta podem variar de acordo com o operador. Por exemplo, Falha, Concluído, Após e Personalizado.

Uma nova porta de saída codificada por cores é exibida na borda do operador.

4. Clique na porta de saída no operador de origem e arraste o link para a porta de entrada do operador de destino.

Um novo link é exibido entre os dois operadores.

5. Repita essas etapas, adicionando links e, se necessário, portas entre os operadores para definir o fluxo do processo.

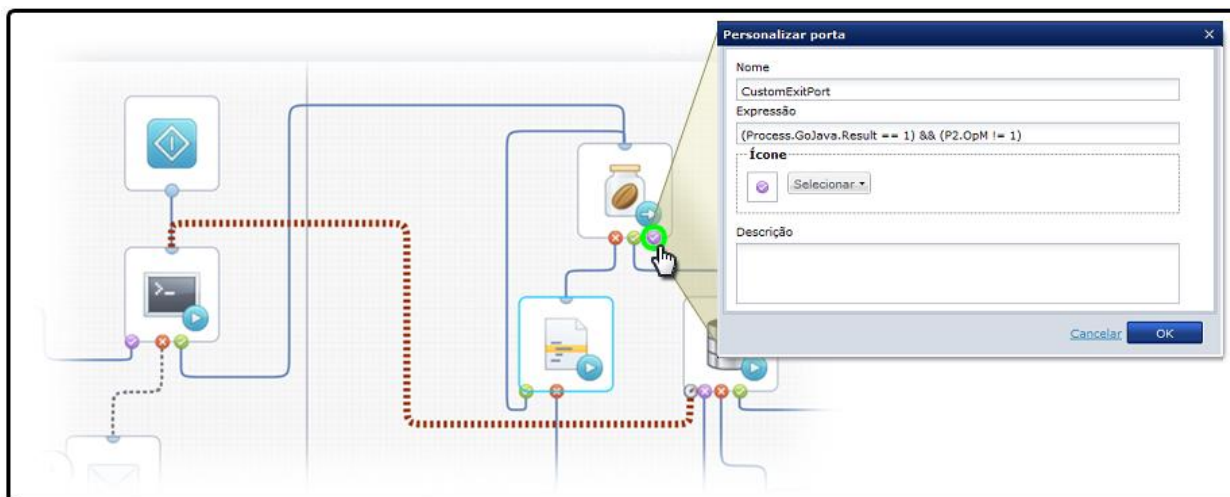
6. Na barra de ferramentas, clique em Salvar.

## Portas de saída e expressões personalizadas

A maioria dos operadores do CA Process Automation oferecem suporte a portas de saída. Ao adicionar uma porta personalizada, especifique o seguinte:

- um nome para a condição
- um ícone para diferenciá-lo das outras portas no operador
- uma expressão booleana que retorne *Verdadeiro* ou *Falso* quando o operador concluir o processamento
- uma descrição opcional

Também é possível mover a porta para melhorar a exibição do link. Pressione a tecla Ctrl enquanto clica na porta para movê-la para posições com suporte ao longo das bordas direita ou inferior do operador.



No campo *Expressão*, digite qualquer expressão JavaScript válida que retorne um valor booleano *Verdadeiro* ou *Falso*. É possível usar a expressão de porta personalizada para avaliar o código de resultado de um operador. O código de resultado indica o resultado do operador e é retornado pela variável de *resultado* no conjunto de dados do operador. O link personalizado somente será ativado se a expressão retorna uma condição *Verdadeira*.

Para ativar um link com base em um valor específico para o código de resultado, use a seguinte sintaxe:

```
Process.Operator.Result == value
```

A palavra-chave *Processo* refere-se ao conjunto de dados do processo. *Operador* é o valor especificado pelo parâmetro *Nome* do operador. *Resultado* é o nome de campo para a variável do código de resultado no conjunto de dados do operador.

Para ativar uma porta, você também pode construir uma expressão como uma comparação entre várias instruções. Neste exemplo, o valor de *GoJava* deve ser 1 e o valor do operador *OpM* no processo *P2* não deve ser 1:

```
(Process.GoJava.Result == 1) && (P2.OpM != 1)
```

Quando um processo é executado e encontra um operador sem nenhuma porta de saída definida, o processo passa para um estado *Bloqueado*.

Quando um operador tem várias portas personalizadas, o aplicativo executa todas as portas com uma condição de saída que retorna um valor *Verdadeiro*. Evite sobrepor a lógica para as portas de saída se não desejar ativar mais de um link de saída de cada vez. É mais provável que essa condição ocorra se você incluir portas padrão e personalizadas no mesmo operador. Por exemplo, se você incluir uma porta personalizada ativada pela expressão *Process.A==5* e uma porta padrão com êxito em um operador, um operador ativar os links de saída quando *Process.A* retornar 5. Para capturar valores e direcionar o processamento para uma única porta, convém usar mais de uma porta de saída personalizada para especificar expressões exclusivas, como:

```
(Process.A == 5) AND (Process.OperatorName.Result == 5)
(Process.A == 5) AND (Process.OperatorName.Result != 5)
(Process.B == "finance") AND (Process.OperatorName.Result == 7)
(Process.OperatorName.Result == 1) AND (Process.A <> 5)
```

Nesse exemplo, também é possível incluir o link de falha padrão para cobrir quando a variável de resultado do operador retornar 0. Se não tiver certeza sobre os diferentes resultados, você poderá usar a função de simulação em um operador para testar os resultados de permutações diferentes de configurações e valores.

**Observação:** use os recursos de assistência de variável (pressione as teclas Ctrl + espaço) para facilitar a construção dessas expressões.

É possível editar um link personalizado a qualquer momento após adicioná-lo a um operador, clicando duas vezes na porta. Se você não pode predefinir um curso de ação para uma determinada condição de saída (por exemplo, quando uma importação de banco de dados falha), é possível omitir um link para ele. Quando uma condição de saída para um operador não é especificada por qualquer link de saída, o processo entra em um estado suspenso até que um usuário possa executar a ação corretiva.

## Quebrar um link para fins de legibilidade

Os links podem se tornar confusos em um processo complicado e atrapalhar a sua visualização. Se você tiver vários links cruzando uma área ou deseja vincular um operador em outras ramificações a um operador distante, poderá interromper um link.

**Observação:** o link em si não é interrompido, somente a representação visual dele. Em vez de uma linha completa desdobrando-se entre dois operadores, a linha é dividida em dois símbolos de link numerados correspondentes nos operadores de origem e de destino.

### Siga estas etapas:

1. Identifique o link que deseja interromper entre um operador de origem e um operador de destino.
2. Clique com o botão direito do mouse no link e selecione Interromper link.

O link interrompido é substituído por marcadores de link correspondentes em ambas as extremidades do link. O link dividido comporta-se da mesma maneira que um link ininterrupto.

**Observação:** para vincular novamente um link interrompido, clique com o botão direito do mouse no marcador de link numerado e clique em Associar link.

## Loops e iterações de processo

Vigilância, monitoramento e outros processos cíclicos são normalmente executados repetidamente. Você pode controlar esses ciclos usando loops. Você pode aplicar um ou mais métodos para executar operadores nos loops:

- Você pode percorrer ou repetir um operador até que alguma condição seja atendida.
- Você pode usar o operador Loop para repetir uma sequência de operadores.
- É possível repetir todo o processo. Um processo em loop pode consistir em vários operadores vinculados.

## Variáveis de sistema para loops

Você pode criar variáveis de loop personalizados e gerenciá-las ou usar as variáveis de sistema disponíveis para os loops incluídos nesta versão do CA Process Automation.

Por exemplo, você poderia criar uma lógica para calcular a duração desde a hora de início da primeira iteração de um loop até a hora atual para cada loop. Você poderia usar até mesmo um código de execução anterior e posterior para configurar a entrada em um loop, como

- inicializar variáveis
- definir contagens de loops
- processar resultados quando o loop for concluído

Em vez de criar e atualizar suas próprias variáveis de contador de loops, você pode aproveitar as variáveis de loop internas. O operador Loop e qualquer outro operador com configurações de loop suportam as seguintes variáveis de conjunto de dados:

- *CurrentLoopIteration*
- *OverallLoopDuration*

Use *CurrentLoopIteration* quando precisar de um contador de loops padrão. A variável *CurrentLoopIteration* contém o valor 0 durante a primeira iteração do loop e é incrementada em 1 no início ou no final de cada iteração adicional. Por exemplo, se o operador for configurado para efetuar loop 3 vezes, no final da execução de todas as iterações, *CurrentLoopIteration* será igual a 3. Especificamente, ela será 0 na primeira iteração, 1 na segunda iteração, 2 na terceira iteração e 3 na última iteração. A última iteração não será executada por violar a condição de loop.

Use *OverallLoopDuration* quando precisar efetuar um loop durante um tempo fixo; por exemplo, efetuar um loop por, no máximo, 5 minutos e, em seguida, sair. Essa variável contém o número de segundos entre o início da primeira iteração do loop e o final da última iteração. *OverallLoopDuration* é atualizada no início e no final de cada iteração do loop. Ela inclui qualquer atraso definido entre as iterações do loop.


**Observação:** não é possível modificar as variáveis de sistema *CurrentLoopIteration* e *OverallLoopDuration*. Embora elas sejam exibidas no conjunto de dados do operador, seus valores não são alterados, a menos que estejam em loop (operador Repetir contagem > 1).

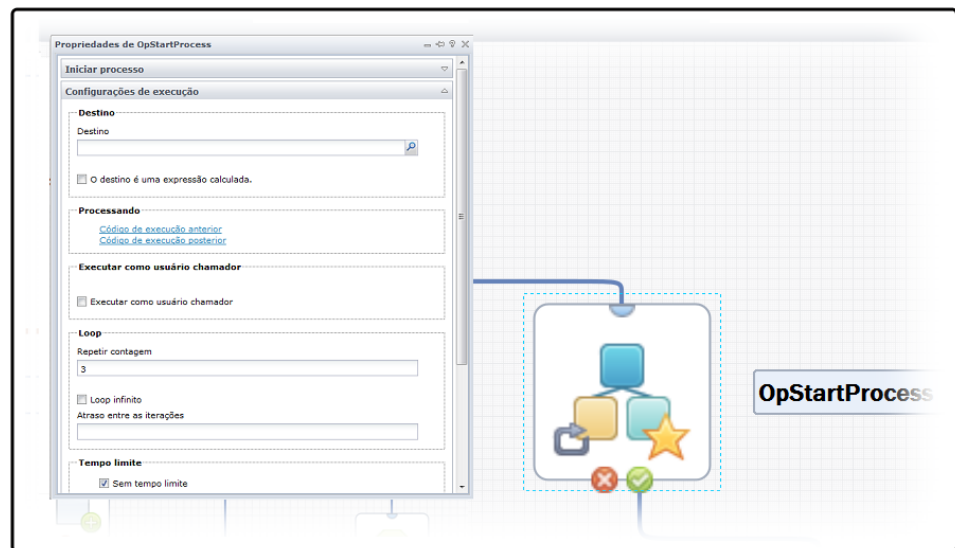
## Loop de um operador em um processo

Um método simples para o loop de um processo consiste em definir os parâmetros de Loop em um operador que ofereça suporte a loop. Abra a paleta Propriedades do operador. Na seção Configurações de execução, no campo Repetir contagem, digite o número de repetições do operador. A opção Repetir contagem é um campo calculado, portanto é possível usar uma variável ou uma expressão para especificar a contagem em tempo de execução. A opção Repetir contagem aceita um número inteiro (o número de vezes de loop) ou um valor booleano (o loop continuará desde que a condição seja avaliada como verdadeira). Exemplos de entradas válidas:

```
3
Process.var < 3
Process.var == false
```

É possível repetir o operador indefinidamente, marcando a caixa de seleção Loop infinito.

O indicador de Loop  é exibido nos operadores que você tenha decidido realizar o loop:



É possível configurar o parâmetro de loop no operador Iniciar processo para executar um processo repetidamente. Isso funciona bem para fazer o loop de algumas iterações de um processo e salvar um instantâneo do histórico do processo para cada loop. No entanto, evite chamar um processo muitas vezes (como em um loop infinito) a partir de outro processo. O CA Process Automation mantém um histórico de todas as instâncias do processo. Chamar um processo em um loop infinito a partir de outro processo poderá usar uma grande quantidade de espaço em disco para salvar dados irrelevantes.

O método preferencial para executar um processo repetidamente consiste em executar um loop cíclico dentro de um processo. Quando necessário, você ainda poderá salvar um instantâneo do histórico de um processo em loop por meio da ramificação no operador Iniciar processo, que inicia uma nova instância desvinculada do processo antes de executar um operador Interromper no final da ramificação.

É possível especificar o período de retenção, ou o tempo para salvar o histórico, nas configurações de diretiva de biblioteca para um orquestrador ou seu touchpoint associado.

**Observação:** se você definir um operador em loop com um tempo limite seguido da ação *Redefinir*, a condição de loop será verificada quando houver movimentação de uma iteração para outra, e não quando ocorrer a redefinição de uma iteração. A variável *OverallLoopDuration* contém o número de segundos desde o início da primeira iteração, incluindo o tempo gasto em todas as iterações de redefinição. As redefinições de iteração do loop não redefinem também *OverallLoopDuration*.

Se você definir um operador para efetuar loop com um tempo limite seguido de uma ação *Continuar*, *OverallLoopDuration* conterà o número de segundos desde o início da primeira iteração até o final da última iteração bem-sucedida. Se o operador expirar, *OverallLoopDuration* não conterà o número de segundos desde o início da primeira iteração até o momento em que o operador expirar.

## Interromper um operador em loop

É possível interromper um operador em loop, adicionando um link de interrupção de outra ramificação do processo.

### Siga estas etapas:

1. Abra e reserve um processo.
2. Crie um link a partir de um operador em uma ramificação independente e anexe-o a um operador em loop.
3. Clique com o botão direito no link e clique no link Parar.

O link é exibido como uma linha vermelha tracejada com um símbolo de interrupção vermelho, próximo ao operador em loop.

4. Clique em Salvar.

Em tempo de execução, a ramificação separada atinge o operador em loop, e as seguintes ações ocorrem:

- O operador em loop é executado e concluído.
- As ações de execução posterior para o operador são executadas.
- O processamento das duas ramificações, agora mescladas, passa para o próximo operador no processo.

## Repita elementos indexados de um campo de conjuntos de dados

Em vez de usar as variáveis internas de sistema *CurrentLoopIteration* e *OverallLoopDuration*, você pode criar variáveis de loop personalizadas e gerenciá-las. As versões anteriores do CA Process Automation exigiam este método. Por exemplo, você poderia criar uma lógica para calcular a duração desde a hora de início da primeira iteração de um loop até a hora atual para cada loop. Você poderia usar até mesmo um código de execução anterior e posterior para configurar a entrada em um loop, como

- inicializar variáveis
- definir contagens de loops
- processar resultados quando o loop for concluído

Para repetir o ciclo de todos os elementos de um campo de conjuntos de dados indexado, use primeiro um operador Executar o JavaScript do grupo Utilitários para inicializar o elemento *CurrentIndex* do campo de conjunto de dados como 0. Por exemplo, a expressão a seguir inicializa o elemento *CurrentIndex* na variável de processo *X* para 0:

```
Process.X.CurrentIndex=0;
```

### Para repetir elementos indexados de um campo de conjunto de dados

1. Conecte o link Saída com êxito do operador Executar o JavaScript ao operador que deseja processar em um loop.
2. Na paleta Conjunto de dados do operador que você deseja repetir, use a configuração do elemento Tamanho para o campo indexado como o valor Repetir contagem na guia Loop. Por exemplo:

```
Process.X.Size.
```

3. Para aumentar a configuração *CurrentIndex* após a conclusão de cada iteração do loop, use uma expressão de código de execução posterior. Por exemplo:

```
Process.X.CurrentIndex=Process.X.CurrentIndex+1;
```

Nesse caso, o elemento *CurrentIndex* é o contador para o loop. É possível usar a configuração *CurrentIndex* para acessar os elementos do campo indexado em expressões calculadas. Por exemplo:

```
Process.X[Process.X.CurrentIndex];
```

**Observação:** se você estiver acostumado com linguagens de programação, como o Visual BASIC, que usam matrizes baseada em um, lembre-se de que um campo de conjunto de dados indexado é uma matriz com base zero. Nas matrizes baseadas em um, o primeiro elemento é indexado por 1, e o último elemento é indexado pelo número de elementos. O primeiro elemento de um campo de conjunto de dados indexado é indexado por 0, e o último elemento é indexado por um menos o valor do elemento Tamanho para o campo.

Além de acessar os elementos em um campo de conjunto de dados indexado, é possível rastrear o número de iteração durante o processamento do loop, para outras finalidades. Use uma variável de processo, como *process.i* para a variável de índice. Em um operador Cálculo do Módulo do intérprete que precede o operador em loop, inicialize a variável de processo para seu valor inicial. Por exemplo, *process.i=1*. Para aumentar a variável de índice após a conclusão de cada iteração do loop, use uma expressão no código de execução posterior para o operador. Por exemplo:

```
Process.i=Process.i+1;
```

**Observação:** as etapas deste tópico não são mais consideradas necessárias; no entanto, elas são incluídas para referência com código herdado. Se as variáveis de sistema mais recentes forem utilizadas, somente a etapa 2 será aplicada e a etapa 3 será substituída, conforme a seguir. Use a variável *CurrentLoopIteration* do operador em loop para acessar os elementos do campo indexado nas expressões calculadas. Por exemplo:

```
Process.X[Process[OpName].CurrentLoopIteration];
```

### Indexar o loop leva em conta outras finalidades

É possível rastrear o número de iteração durante o processamento do loop para acessar elementos de um campo de conjunto de dados indexado, ou para outras finalidades. Use uma variável de processo para a variável de índice (por exemplo, *process.i*). Em um operador Cálculo do módulo do Intérprete que precede o operador em loop, inicialize a variável de processo com seu valor inicial para o loop:

```
Process.i=1;
```

Para aumentar a variável de índice após a conclusão de cada iteração do loop, use uma expressão na guia Ações da Execução posterior das propriedades do operador. Por exemplo:

```
Process.i=Process.i+1;
```

**Observação:** este tópico não é mais considerado necessário; no entanto, ele é incluído para referência ao trabalhar com código herdado.

### Erros e exceções de loop

Se ocorrer um erro em relação ao código de execução anterior ou posterior:

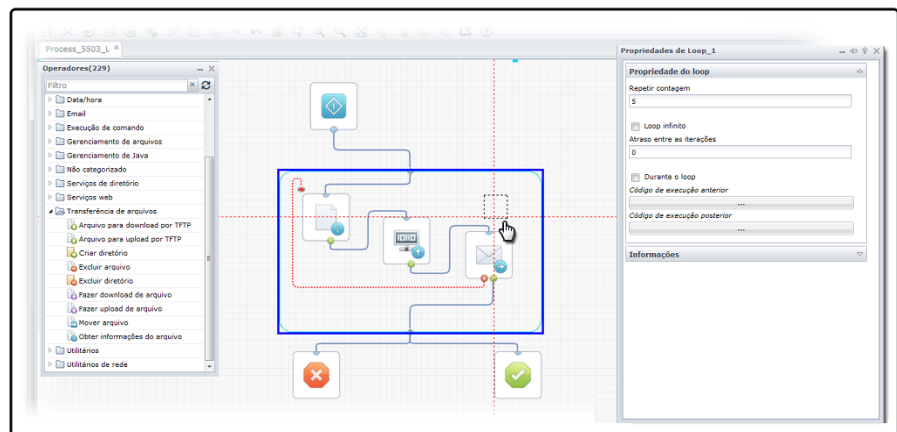
- Os logs do processo indicarão que ocorreu falha na execução da condição anterior ou posterior do operador Loop.
- O manipulador de exceção de processo é disparado com uma exceção *cancelar*. A *Origem* e *SourceROID* apontam para o operador Loop.

## Loop de uma série de operadores

Também é possível executar o loop de uma série de operadores em um operador Loop.

**Siga estas etapas:**

1. Abra um processo.
2. Arraste o operador Loop do grupo de operadores Padrão para o processo.
3. Redimensione a caixa delimitadora para o operador Loop a fim de acomodar a sequência de operadores que deseja adicionar ao loop.
4. Arraste os operadores para a caixa.  
A cor da caixa delimitadora muda para indicar que é o destino dos operadores arrastados.
5. Adicione portas e links.
6. Vincule o ponto de entrada na caixa ao primeiro ícone na ramificação e vincule o último ícone à porta de saída na caixa.
7. Adicione links para o operador Loop e a partir dele:
  - a. Vincule pelo menos um operador fora da caixa com a porta de entrada.
  - b. Vincule a porta de saída a, pelo menos, um operador fora da caixa.



8. Clique duas vezes no operador Loop para exibir suas propriedades.

A paleta Propriedades do operador Loop é exibida. Se ela não for exibida, selecione Propriedades no menu Exibir.

9. Defina as propriedades do loop e clique em Aplicar.

**Observação:** é possível criar uma ramificação de loop infinito, marcando a caixa de seleção Loop infinito. Assim como em qualquer operador de loop infinito, você pode interromper o loop, adicionando um link de interrupção por meio de outro operador.

**Observação:** os valores das variáveis do sistema *CurrentLoopIteration* e *OverallLoopDuration* são atualizados para cada instância do processo e são exibidos no conjunto de dados. Mesmo que Repetir contagem para o operador Loop esteja definido como 1, *CurrentLoopIteration* mostrará 1 e *OverallLoopDuration* mostrará 10 no final de uma única instância. Por outro lado, as variáveis do sistema *CurrentLoopIteration* e *OverallLoopDuration* apenas serão atualizadas para outros operadores quando Repetir contagem for maior que 1. Isso ocorre porque o operador Loop *sempre* entra em loop, mesmo que seja uma única vez.

**Mais informações:**

[Interromper um operador em loop](#) (na página 118)

## While e Do While Loops

Use a caixa de seleção Durante o loop do operador Loop para gerenciar o comportamento do loop. Quando marcado, o operador Loop comporta-se como um *while* loop. Quando desmarcado, o operador Loop comporta-se como um *do while* loop.

Qualquer operador Loop existente no conteúdo desenvolvido antes de uma atualização para o CA Process Automation 4.0 terá o campo Durante o loop desmarcado. Eles continuarão se comportando como *do while* loops. Você pode transformar esses operadores Loop em *while* loops, marcando a caixa Durante o loop associada.

As duas diferenças principais entre os operadores Loop *while* e *do while* são:

- Os *while* loops verificam a condição de loop especificada no campo Repetir contagem antes de ela executar a primeira iteração e cada iteração subsequente.
- Os *do while* loops verificam a condição de loop especificada no campo Repetir contagem, no final de cada iteração, portanto, é garantido a execução de pelo menos a primeira iteração do loop.

Observação: um operador Loop pode ser definido para se comportar como um *while* ou um *do while* loop. Os outros operadores que oferecem suporte a loop só podem se comportar como *do while* loops.

## A sequência lógica de um operador Loop

Os campos Código de execução anterior e Código de execução posterior estão disponíveis nas propriedades do operador Loop. É possível inserir um código JavaScript nesses campos para ser executado com cada iteração do loop. O CA Process Automation executa qualquer código de execução anterior e posterior para um operador Loop para cada iteração do loop.

**Observação:** no conteúdo desenvolvido antes de uma atualização para o CA Process Automation 4.0, os operadores de loop terão os campos Código de execução anterior e Código de execução posterior vazios.

A sequência de processamento de qualquer condição anterior e posterior depende do tipo de loop.

### Operadores While loop:

1. Execute a condição anterior.
2. Verifique a condição *while* loop, conforme indicado pelo campo Repetir contagem.
  - Se a condição de loop for bem-sucedida:
    - a. Redefina os operadores dentro do operador Loop.
    - b. Execute os operadores dentro do operador Loop.
    - c. Execute a condição posterior no final da iteração do Loop.
    - d. Efetue um loop reverso até a etapa 1.
  - Se a condição de loop falhar:

Saia do operador Loop sem executar a condição posterior e, em seguida, execute o próximo operador após o operador de loop.

### Operadores Do while loop:

1. Redefina os operadores dentro do operador Loop.
2. Execute a condição anterior.
3. Execute os operadores dentro do operador Loop.
4. Execute a condição posterior.
5. Verifique a condição *do while* loop, conforme indicado pelo campo Repetir contagem.
  - Se a condição de loop for bem-sucedida, efetue um loop de volta para a etapa 1.
  - Se a condição de loop falhar, saia do operador Loop e execute o próximo operador após o operador Loop.

## Loop de um processo

Para executar um processo várias vezes, crie uma ramificação cíclica em um objeto de processo, em vez de usar o operador Iniciar processo para chamar um processo repetidamente por meio de outro processo. Esse método envolve o loop de um processo cíclico estabelecendo dois operadores Iniciar. Um operador vai para o início e o outro para o fim.

### Siga estas etapas:

1. Abra e reserve um processo.
2. Coloque os seguintes operadores:
  - a. Iniciar
  - b. Interrupção bem-sucedida
  - c. Interrupção com falha
  - d. Uma série de operadores que você deseja repetir.
3. Adicione um operador Iniciar no ponto da sequência em que ele termina e onde você deseja reiniciar a ramificação.
4. Vincule o último operador da ramificação ao segundo operador Iniciar. O operador Iniciar tem um link de entrada que permite que ele seja colocado no final de uma sequência de etapas.

**Observação:** quando a sequência de processamento chega ao segundo operador Iniciar mostrado na figura, ela reinicializa todos os operadores e reinicia o processo.

5. Para interromper esse tipo de processo cíclico, incorpore a lógica que lidera a ramificação cíclica. Adicione uma porta de saída em um operador que leve diretamente a um operador Interromper ou a uma ramificação diferente.

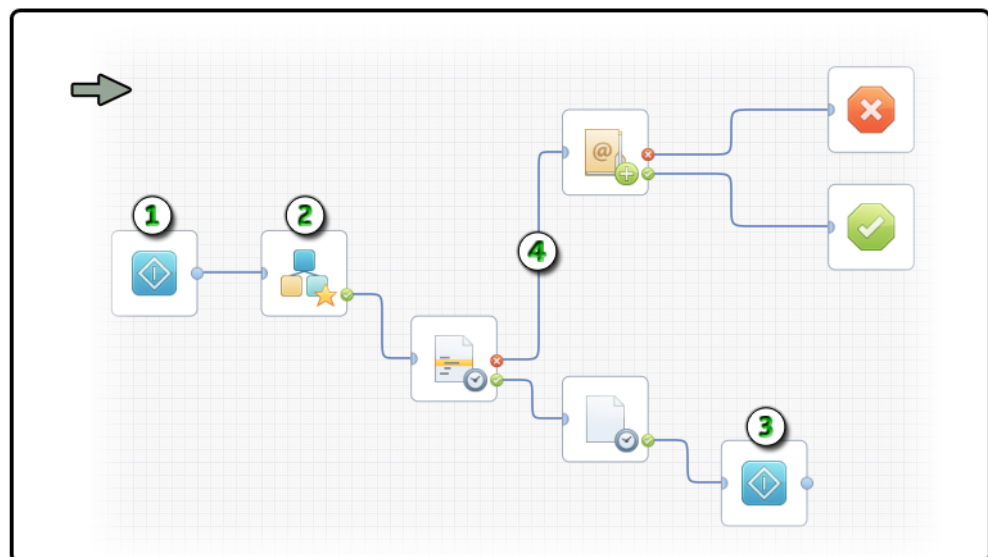
## Daemons e outros processos de loop

Quando desejar executar um processo repetidamente, crie uma ramificação cíclica em um objeto de processo, em vez de usar o operador Iniciar processo para chamar um segundo processo repetidamente. Um processo cíclico é executado em um loop infinito. Cada iteração não cria novas instâncias do objeto de processo. A principal vantagem da implementação de uma ramificação cíclica é que a lógica que sai do loop está no próprio processo. O processo não depende de fatores externos em um processo pai. O processo é executado como um objeto autônomo.

Os exemplos de processos de loop executados indefinidamente incluem:

- Processos de loop contínuos para monitorar o uso do sistema ou da rede
- Sondagem do estado de redes ou componentes do sistema
- Organização e verificação de eventos ou mensagens
- Daemons
- Serviços

### Exemplo de processo de loop



**Item:**    **Descrição:**

- |          |  |
|----------|--|
| <b>1</b> | <b>Operador Iniciar 1:</b> inicie o processo com um operador Iniciar padrão.   |
| <b>2</b> | <b>Operador Iniciar processo:</b> crie uma sequência de operadores que representem a funcionalidade que você deseja repetir ou da qual deseja efetuar um loop. Essa ramificação pode até mesmo incluir um operador Iniciar processo para iniciar as instâncias de um segundo processo. |

Item:	Descrição:
3	<b>Operador Iniciar 2:</b> o operador Iniciar tem um link de entrada opcional. Quando o processo fluir para o segundo operador Iniciar, ele reinicializará todos os outros operadores e reiniciará o processo novamente no Operador Iniciar 1.
4	<b>Porta de saída e link:</b> para sair desta sequência em loop, especifique uma condição de saída.

**Observação:** um método alternativo para interromper um processo cíclico consiste em usar uma ramificação independente com seu próprio operador Iniciar. A ramificação pode aguardar a parte de loop do processo para alterar uma variável ou liberar um recurso. Quando esses eventos ocorrerem, a ramificação independente será executada; por exemplo, enviando um alerta de email e processar um operador Interromper.

## Controle de processo

Em geral, os processos complexos necessitam de uma hierarquia de subprocessos. Na programação tradicional, a funcionalidade é modular. As equipes separam a complexidade em procedimentos, métodos ou bibliotecas que podem ser usadas repetidamente. De maneira semelhante, no CA Process Automation, há métodos para simplificar processos complexos. Você pode criar modelos de processos subordinados como ramificações ou como processos separados. Um processo subordinado ou *filho* separado pode ser chamado por um processo *pai* por meio de um operador Iniciar processo. Por exemplo, um processo pai que está gerenciando os sistemas de computador de um banco pode incorporar os seguintes processos filho usando os operadores Iniciar processo:

- Durante cada dia útil, processa automaticamente os pagamentos de faturas.
- Durante cada noite, realiza um backup diário.
- No final de cada dia, processa e faz verificações.
- No final de cada mês, calcula os juros das contas.

Os processos subordinados definem lógica ou fisicamente os processos filho distintos dentro de um sistema. A estrutura de um processo pai sincroniza processos filho, gerencia os recursos compartilhados entre os processos e define o gerenciamento de erros e as dependências entre os processos em uma escala que abrange todo o sistema.

## Processos filho

Use o operador Iniciar processo para iniciar um processo secundário ou *filho* a partir de uma instância em execução de um processo primário ou *pai*. O operador Iniciar processo inicia uma nova *instância* de processo em um touchpoint especificado. O operador Iniciar processo inicia as variáveis de processo associadas à nova instância.

O operador Iniciar processo pode iniciar uma nova instância de um processo no modo *vinculado*, *desvinculado* ou como um processo *embutido*:

- **Modo vinculado:** o processo que inicia a nova instância é conhecido como processo *pai*. A nova instância é chamada de processo *filho*. O aplicativo conclui o processamento de toda a nova instância. Um processo filho que termina seu fluxo no modo vinculado pode copiar seus valores de conjunto de dados para o conjunto de dados no processo de chamada pai.

Se um processo pai ativar um operador Interromper em uma ramificação antes de um processo filho ativar seu próprio operador Interromper, é possível que a instância pai seja concluída *antes* da instância filho.

- **Modo desvinculado:** o operador Iniciar processo inicia uma nova instância de processo filho que age como se ele não tivesse pai. No modo desvinculado, o fluxo de trabalho que inicia uma nova instância de outro processo é imediatamente concluído após enfileirar a solicitação inicial. Um processo que termina no modo desvinculado não pode copiar seus valores de conjunto de dados para o conjunto de dados no processo de chamada pai.
- **Processo embutido:** o processo filho é executado como uma instância separada. Um processo pai limitou o controle sobre o processo filho. O processo filho embutido é totalmente vinculado ao pai com acesso ao contexto e ao ciclo de vida do pai, como instanciação e arquivamento. Não é possível executar um processo embutido no modo desvinculado.

## Configurar um processo filho

Configure o operador Iniciar processo em um processo pai para controlar o comportamento do processo filho que está sendo iniciado.

### Siga estas etapas:

1. Abra e reserve um processo no Criador de processos.
2. Na paleta Operadores, expanda Controle de processo e arraste um operador Iniciar processo para o seu processo.
3. Clique duas vezes no operador Iniciar processo para exibir suas propriedades na paleta Propriedades.
4. Expanda o grupo Iniciar processo.
5. No campo Nome do processo, digite o caminho completo do processo.
6. No campo Código de inicialização do conjunto de dados do processo, clique no botão Procurar, indicado por ... (reticências) para abrir toda a caixa de diálogo de script.
7. Digite as instruções JavaScript para inicializar as variáveis no conjunto de dados do processo filho que você está iniciando. Nesse contexto, a palavra-chave *Processo* se refere ao conjunto de dados do processo da nova instância que está sendo iniciada. A palavra-chave *Chamador* se refere ao conjunto de dados da instância pai que contém o operador Iniciar processo. Este é o único contexto no qual a palavra-chave do chamador está disponível. Esse contexto também é o único no qual a palavra-chave *Processo* não faz referência ao processo que contém um operador.
  - Processo pai: Processo A
  - Processo filho: Processo B
  - Iniciar o operador Processo A inicia o Processo B filho
  - A inicialização do conjunto de dados local B ocorre com o conjunto de dados A como o *Chamador*

As palavras-chave *Processo* e *Chamador* são obrigatórias para fazer referência às variáveis do conjunto de dados do processo pai ou filho. Se você omitir as duas palavras-chave em um nome de variável, o aplicativo procurará uma variável com escopo de cálculo. Ele não verificará variáveis com nomes semelhantes no conjunto de dados pai ou filho. Por exemplo, o código a seguir falhará se nenhuma variável *X* com escopo de cálculo tiver sido criada anteriormente na caixa de diálogo de script local:

```
Process.X = Caller.X;  
Process.Y = X + 100;
```

8. Clique em Salvar para fechar a caixa de diálogo Código de inicialização.
9. No campo Modo, selecione Vinculado, Desvinculado ou Embutido.

Se você selecionar Desvinculado, o campo Data de início será ativado. Ele especifica a data em que a instância desvinculada do processo é iniciada. O valor padrão é a data em que o operador é executado, o que é indicado como *System.Date*. Da mesma forma, o campo Hora de início especifica o horário em que a instância desvinculada é iniciada. O valor padrão é *System.Time*.

**Observação:** as versões anteriores deste aplicativo incluíam operadores Executar processo separados para os modos vinculado e desvinculado. Os dois operadores executavam, na verdade, a mesma função. A única diferença era que, para o operador posicionado junto a Executar processo, a caixa de seleção Desvincular após o início ou após a solicitação de enfileiramento era inicialmente desmarcada. Para o operador Executar processo desvinculado, a caixa de seleção era inicialmente marcada. Em qualquer um dos casos, era possível marcar ou desmarcar a caixa de seleção Desvincular após o início ou após a solicitação de enfileiramento a qualquer momento depois que você tivesse incluído o operador Executar processo para alterar seu modo de início. Nesta versão, todos esses operadores são importados como operadores Iniciar processo.

## Iniciar variáveis de processo filho com as palavras-chave Chamador e Processo

Para inserir um script para iniciar variáveis do processo no processo filho, clique no botão de navegação Código de inicialização do conjunto de dados do processo para abrir a caixa de diálogo de script do processo. Na caixa de diálogo de script do processo, é possível inserir as instruções JavaScript para inicializar as variáveis no conjunto de dados do processo que você está iniciando.

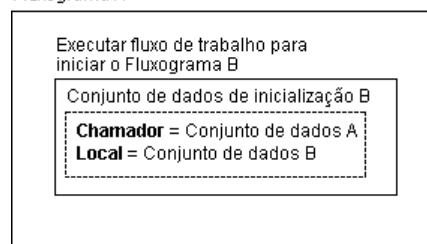
### Palavra-chave Processo

Refere-se ao conjunto de dados do processo da nova instância que está sendo iniciada. Esse contexto é o único no qual a palavra-chave Processo não faz referência ao processo que contém um operador.

### Palavra-chave Chamador

Refere-se ao conjunto de dados da instância pai que contém o operador Executar processo. Este é o único contexto no qual a palavra-chave do chamador está disponível.

Fluxograma A



Na caixa de diálogo Script de processos, as palavras-chave do chamador e do processo são obrigatórias para se referir às variáveis do conjunto de dados do processo pai ou filho. Se você omitir palavras-chave em um nome de variável na caixa de diálogo Script de processo, o Módulo do intérprete procurará uma variável de cálculo de escopo. Os Módulos do intérprete não procurarão por uma variável com nome semelhante no conjunto de dados pai ou filho. Por exemplo, o seguinte código falhará se nenhuma variável com cálculo de escopo X foi criada anteriormente na caixa de diálogo de script local:

```
Process.X = Caller.X;  
Process.Y = X + 100;
```

## Retornar o conjunto de variáveis para o processo pai

O conjunto de dados para um processo filho concluído pode ser acessado usando uma expressão no processo pai. O nome do operador Iniciar processo faz referência ao conjunto de dados do processo filho no conjunto de dados local do processo pai. O código a seguir no processo pai faz referência a uma variável de conjunto de dados de um processo filho:

```
Process.Operator_Name.Field_Name
```

*Operator\_Name* representa o nome do processo Iniciar processo no processo pai, e *Field\_Name* é a variável de conjunto de dados que você deseja acessar no processo filho.

Isso funciona apenas para processos iniciados no modo vinculado. Os processos iniciados no modo desvinculado tornam-se o processo raiz em uma sequência de chamadas e não copiam seus conjuntos de dados para o processo que os iniciou.

## Iniciar processos recursivamente

Diferentemente do uso da memória, não há restrições quanto ao número de processos que podem ser iniciados em uma cadeia. Um processo pode, até mesmo, iniciar outra instância de si mesmo recursivamente.

Evite chamar processos recursivamente no modo vinculado, pois isso pode resultar em uma cadeia de chamadas infinita. Entretanto, muitas vezes é útil para um processo iniciar outra instância de si mesmo no modo desvinculado. Por exemplo, é possível salvar uma imagem de um processo de monitoramento em um determinado estado e continuar executando o processo. Nesse caso, o processo de monitoramento pode iniciar uma nova instância de si mesmo no modo desvinculado e, em seguida, executar um operador Interromper para ser encerrado. Em seguida, a instância encerrada é salva e um administrador pode examiná-la em seu estado pré-encerrado.

## Processo embutido

O operador Iniciar processo é usado para chamar processos filho. O processo filho chamado é executado como uma instância separada. Um processo pai tem controle limitado sobre o processo filho e pode resultar em uma sobrecarga de desempenho no tempo de execução.

O modo embutido permite que você execute um processo filho e expanda-o no processo pai. Um processo filho embutido tem acesso ao contexto do pai. O ciclo de vida do filho embutido, incluindo instanciação, arquivamento etc., é totalmente vinculado ao pai.

O CA Process Automation permite que os usuários decidam se desejam executar um processo como um processo embutido.

## Configurar um processo embutido

Você pode configurar um operador de processo para executar um processo filho no modo embutido.

### Siga estas etapas:

1. No operador Iniciar processo, paleta Propriedades, clique no botão Navegador de objetos (...).  
O Navegador de objetos é exibido.
2. Selecione o processo filho.
3. No campo Modo, selecione *Embutido*.

O processo filho está configurado para ser executado como um processo embutido.

4. Marque a caixa de seleção Herdar manipulador de alteração de rota do processo pai.

Os manipuladores de alteração de rota do processo pai são carregados para o processo filho.

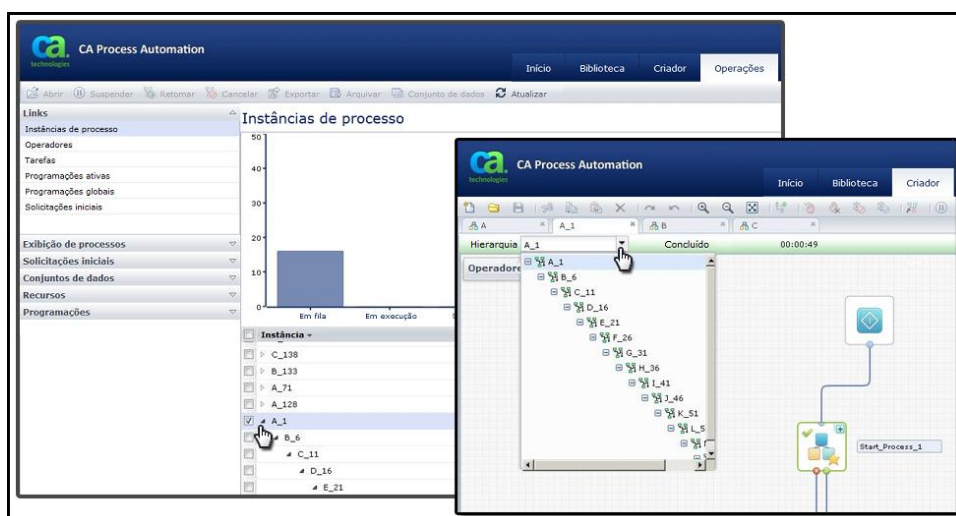
**Observação:** a caixa de seleção Herdar manipulador de alteração de rota do processo pai é ativada somente para processos filho embutidos. O CA Process Automation avalia o processo filho quando o processo pai é iniciado. O processo filho embutido é carregado e expandido no processo pai.

## Exibir um processo filho embutido

É possível exibir um processo filho embutido em tempo de execução usando um dos métodos a seguir:

- No Criador, clique no ícone de adição no lado superior direito do operador Iniciar processo da instância pai.
- No Criador, clique na lista suspensa Hierarquia e selecione um processo filho listado sob o pai.
- Na página Operações, clique em uma instância pai.

O gráfico a seguir mostra os três métodos.



## Rotas de processo

As rotas em um processo, geralmente chamadas de *diagramas de raia*, fornecem um meio de dividir o processo em partes lógicas diferentes. É possível adicionar, inserir, redimensionar e remover diagramas de raia usando o Criador de processos. É possível definir regras que são disparadas quando os links entre as etapas de um processo ultrapassam um limite entre as rotas. As regras definem etapas adicionais a serem executadas quando o processamento ultrapassa os limites dos diagramas de raia.

Você pode organizar visualmente os operadores de processo em um ou mais diagramas de raia. Linhas paralelas mostram os limites da rota. O padrão de orientação da rota de um novo processo é vertical. Inicialmente, um novo processo tem uma única rota, denominada *Rota\_0*. As rotas adicionais são nomeadas em sequência à medida que são adicionadas, por exemplo, *Rota\_1*, *Rota\_2* e assim por diante. Você pode personalizar os nomes para organizar as ramificações de um processo de maneira significativa.

Quando houver várias rotas em um processo, você poderá definir regras de alteração de rota. Essas regras especificam sequências de operadores que são chamadas apenas quando a execução de um processo ultrapassa duas rotas. Uma regra de alteração de rota é semelhante a qualquer sequência de operadores na guia Editor principal, exceto pelo fato de que cada regra é iniciada com um operador Alterar a rota, em vez do operador Iniciar. As propriedades no operador Alterar a rota definem a transição entre as rotas de origem e destino, que chamam a sequência de operadores vinculados ao operador Alterar a rota.

É possível adicionar qualquer número de rotas na guia Editor principal e, em seguida, criar links que ultrapassem os limites de rota entre os operadores no processo. Na guia Manipulador de alteração de rota, defina as sequências de operadores chamados quando a execução ultrapassa limites. Quando a execução de um processo ultrapassa um limite de rota, ela chama um processo no manipulador de alteração de rota, definido para ocorrer para a transição específica (por exemplo, da rota A para a rota B).

Se você definir mais de uma regra para determinada transição, o aplicativo usará a primeira regra existente na seguinte ordem de prioridade:

Prioridade	Correspondências e execuções
1	Regra definida para rotas específicas de origem e de destino (A para B)
2	Regra definida para uma rota de origem específica (A para <i>qualquer</i> )
3	Regra definida para uma rota de destino específica ( <i>qualquer</i> para B)
4	Regra definida por qualquer uma das duas rotas ( <i>qualquer</i> para <i>qualquer</i> )
5	Nenhuma regra, se nenhuma das correspondências listadas acima existir.

Se nenhuma regra de alteração de rota for definida no objeto de processo para a instância atual de um processo, o aplicativo procurará uma regra de alteração de rota correspondente no objeto de processo padrão, especificado nas configurações de propriedade do orquestrador. Isso somente ocorrerá se nenhuma regra de alteração de rota for definida no objeto de processo atual. O aplicativo não verificará o objeto de processo padrão se houver qualquer regra de alteração de rota definida no objeto de processo atual, mesmo quando não ocorrer correspondência alguma.

Quando o processamento ultrapassa uma transição entre as rotas, o aplicativo:

- Suspende o processo depois de concluir os operadores em execução no momento.
- Tenta corresponder a uma regra definida no objeto de processo ou no processo padrão para o orquestrador, na seguinte ordem:

Regras de alteração de rota	Correspondências	Ação
São definidas no objeto de processo	Uma regra no objeto de processo na ordem de prioridade listada acima	Execute a regra definida no objeto de processo.
Não estão definidas no objeto de processo	Uma regra no objeto de processo padrão para o orquestrador na ordem de prioridade listada acima.	Execute a regra definida no objeto de processo padrão.
	Nenhuma correspondência.	Ignorar a transição.

Em seguida, o aplicativo continua processando o processo.

## Criar pistas horizontais ou verticais

Os diagramas de raia são orientados horizontal ou verticalmente, conforme configurado nas propriedades do processo. Selecione a orientação dos diagramas de raia antes de iniciar a adição de operadores a um processo. Configure a orientação antes de adicionar outros diagramas de raia a um processo, já que não é possível alterar a orientação de vários diagramas de raia em um processo.

### Siga estas etapas:

1. Abra e reserve um processo.
2. No Criador de processos, clique em Propriedades.
3. Na paleta Propriedades do processo, expanda a seção Configuração.
4. Em Orientação de rota, clique em Horizontal ou Vertical.
5. Clique em Aplicar.

## Gerenciar raias

É possível adicionar, inserir, redimensionar e remover diagramas de raia usando o Criador de processos.

### **Para adicionar uma nova rota à direita das rotas verticais ou à parte inferior das rotas horizontais:**

1. No painel principal do Editor do Criador de processos, clique com o botão direito do mouse no painel de processo.
2. Clique em pistas e em Adicionar rota.

**Observação:** como alternativa, é possível clicar no botão Adicionar rota na barra de ferramentas.

Uma nova rota é adicionada ao painel principal do editor.

### **Para inserir uma rota em qualquer outro local na guia Editor principal:**

1. No painel principal do Editor do Criador de processos, clique com o botão direito do mouse na rota adjacente à qual você deseja adicionar a nova rota.
2. Clique em Rotas, Inserir rota, clique em Inserir rota à esquerda ou Inserir rota à direita.

Uma nova rota é inserida.

3. Se as rotas têm orientação horizontal, clique em Inserir rota acima ou Inserir rota abaixo

Uma rota é inserida acima ou abaixo da rota existente de acordo com a seleção feita.

4. Para redimensionar uma rota existente, clique na linha separadora entre as rotas e arraste-a para a esquerda ou para a direita, para rotas orientadas verticalmente, ou para cima ou para baixo, para rotas orientadas horizontalmente.

**Observação:** a largura mínima de um diagrama de raia é 50 pixels.

É possível remover uma rota de um processo mesclando-a com uma rota existente. Ao mesclar uma rota, é possível mesclá-la com uma rota à esquerda ou à direita de uma rota orientada verticalmente ou acima ou abaixo de uma rota orientada horizontalmente.

### **Para remover uma rota de um processo**

1. No painel principal do Editor do Criador de processos, clique com o botão direito do mouse na rota que deseja excluir.
2. Clique em Rotas, em Mesclar rotas e em Mesclar com o lado esquerdo ou Mesclar com o lado direito.

**Observação:** se as rotas forem orientadas horizontalmente, clique em Mesclar rota acima ou Mesclar rota abaixo para mesclar.

## Regras de manipulação de rotas

Quando um processo contém várias rotas, você pode definir regras de alteração de rotas. Essas regras especificam sequências de operadores que são chamadas apenas quando a execução de um processo ultrapassa um limite de rota. Cada regra é definida para ocorrer em uma das seguintes transições de rota:

- A partir de uma rota específica para outra rota específica (da rota A para a rota B)
- De uma rota específica para qualquer outra rota (rota A para *qualquer*)
- De qualquer rota para uma rota específica (*qualquer* para rota B)
- De qualquer rota para qualquer outra rota (*qualquer* para *qualquer*)

As regras de alteração de rota se parecem com qualquer sequência de operadores na guia Editor principal, exceto pelo fato de que cada regra começa com um operador de alteração de rota em vez de um operador inicial. As propriedades do operador de alteração de rota definem a transição entre as rotas de origem e destino, que chamam a sequência de operadores vinculada ao operador de alteração de rota.

### Siga estas etapas:

1. Na parte inferior do Criador de processos, selecione a guia Manipulador de mudança de rota.
2. No grupo Padrão da paleta Operadores, arraste um operador Alterar rota para o painel Manipulador de mudança de rota.
3. Clique duas vezes no ícone Alterar rota e clique no painel de propriedades Alterar rota do painel Propriedades.
4. No campo Nome, digite um nome para identificar a regra.
5. Para origem e destino, selecione a combinação de rotas que acionam a regra.
6. Clique em Aplicar.
7. Na paleta, arraste mais operadores para o processo que conclui a regra para a transição e configure cada um conforme necessário.
8. Na barra de ferramentas, clique no botão Salvar.

A nova regra de transição está concluída.

## Versões de processos

O CA Process Automation sempre executa a cópia disponibilizada da versão atual de um objeto de processo. Quando o orquestrador começa a executar a versão atual de um objeto de processo, ele cria uma cópia dessa versão do processo na biblioteca de automação. O sistema processa os operadores em uma instância e cria ou faz referência às variáveis do conjunto de dados do processo na instância. As alterações em uma instância de um processo não afetam a definição básica do processo. Definições básicas são acessadas usando-se o navegador da biblioteca. É possível exibir ou editar a definição básica e as instâncias de uma versão de um processo usando o Criador de processos.

## Documentar um processo

É possível gerar a documentação do processo. A documentação do processo pode incluir um gráfico da criação do processo, conforme ele é exibido na tela, na guia Criador.

Use comentários, seus próprios nomes de operador e rótulos de objetos para complementar a documentação do processo gerada. Comentários, nomenclatura e rótulos apropriados ajudam outros criadores a compreender o que seu processo faz e como ele é construído. A inserção de comentários em um processo fornece detalhes sobre cadeias, processos ou regiões em um processo. Os comentários permanecem estacionários em um processo. Não use comentários para identificar operadores específicos, pois os operadores geralmente são realocados no espaço de trabalho para se ajustarem aos links e a outros operadores. Em vez disso, use a propriedade Nome nas propriedades de informações do operador específico para identificá-los.

### **Mais informações:**

[Gerando e usando a documentação do processo](#) (na página 449)

## Adicionar comentários a um processo

Use o operador Comentário do grupo Padrão da paleta Operadores para adicionar comentários em um processo. Você pode alterar a sequência de caracteres da propriedade Nome do operador. Os comentários são importantes para documentar as etapas de um processo e permitir mais espaço do que os rótulos.

### Para adicionar um comentário a um processo

1. Arraste um operador Comentário do grupo Padrão para o processo.  
Um objeto Comentário é exibido com o nome padrão *Comentário*.
2. Clique duas vezes no comentário para abrir a caixa de diálogo Propriedades do comentário.
3. Substitua o texto inicial do comentário pelo texto que deseja exibir na tela.  
Você pode alterar a cor do plano de fundo, o formato do texto e o alinhamento.
4. Clique em OK para aplicar suas alterações.
5. Clique em Salvar para salvar a criação do processo.

## Defina o nome de um operador em um processo

A propriedade *Nome* do operador identifica um operador colocado em um processo. As expressões usam o nome para acessar o conjunto de dados do operador no formato:

```
Process.operator_name.field_name
```

Por padrão, a propriedade *Nome* também é usada para identificar um operador no Criador de processos quando você ativa a opção Informações do ícone para um processo.

Quando você adicionar um operador ao processo, um nome padrão será gerado, indicando a tarefa executada pelo operador. É possível alterar esse texto a fim de fornecer informações mais significativas ou específicas sobre o operador, referentes ao sistema.

### Para alterar o nome de um operador

1. Clique duas vezes em um operador em um processo ou clique no operador e selecione Propriedades no menu Exibir.
2. Nas propriedades de informações, no painel Propriedades, no campo Nome, digite uma breve descrição para o operador.

Os nomes de ícones podem ser compostos por caracteres alfanuméricos e por sublinhado (\_).

## Alterar e exibir informações do operador em um processo

O nome é incluído em um campo de texto que identifica opcionalmente um operador em um processo. A propriedade Texto do operador define esse campo. A opção Informações do operador, no menu Exibir, oculta ou exibe esse campo de texto ao lado dos operadores em um processo.

Para gerar um valor padrão para a propriedade Texto, o CA Process Automation combina as configurações de Nome do operador e Parâmetro do operador.

### Siga estas etapas:

1. Clique duas vezes em um operador em um processo.
2. Nas propriedades de Informações no campo Nome da paleta Propriedades, digite o texto a ser exibido para o operador no processo.

**Observação:** use as outras configurações da guia Informações do painel Propriedades para:

- Especificar um ícone personalizado em vez do ícone padrão do operador.
- Substituir as preferências do objeto para texto atualizado automática ou manualmente que é exibido com o operador.
- Definir a posição do texto, a cor do plano de fundo, a fonte e o alinhamento.

## Conteúdo Self-Contained

Você pode incorporar uma criação de processo, uma instância de tempo de execução, um objeto de automação ou uma exibição específica a outros produtos e interfaces dinâmicas usando URLs designados. Essa integração oferece as funcionalidades a seguir:

- Outras equipes de desenvolvimento podem adotar produtos e soluções que utilizam o CA Process Automation e que oferecem exibições integradas em uma página de uma janela. Por exemplo, como um cliente, administrador ou provedor de serviços, é possível integrar partes específicas da interface de usuário do CA Process Automation nos portais existentes, como sites, intranets e repositórios do Sharepoint.
- Como criador de processos, você pode construir um fluxo de trabalho de processo que envia emails com um link direto para uma tarefa específica, de forma que a tarefa possa ser aprovada de maneira direta.
- O conteúdo independente facilitado por links de atalho para exibições específicas economiza tempo ao acessar informações importantes e relevantes.

## Links de conteúdo independente

Os links a seguir oferecem suporte a conteúdos independentes em outras exibições, outras estruturas, outros web parts ou outros portais. Você também pode enviar emails com esses links para facilitar as exibições diretas em objetos específicos.

Em vez de [URL do servidor], use o URL de sua implantação do CA Process Automation:

```
http://server:port/itpam/Web.jsp
```

Em vez de <caminho>, especifique o caminho absoluto para o objeto de automação e seu nome no Navegador da biblioteca:

```
/MyProjectFolder/Folder_1/MyStartForm
```

**Observação:** esses links diferenciam maiúsculas de minúsculas.

### Objetos de automação:

Os objetos do CA Process Automation são endereçáveis por URL. Esses URL separados podem ser incorporados dentro de iframe ou portlets de outros produtos. Por exemplo, é possível disparar um processo e ter a interface de usuário de acompanhamento dessa instância incorporada a um portlet personalizado.

#### Formulário de solicitação de interação

```
[URL do servidor]?page=Form&refPath=<caminho>
```

#### Formulário de solicitação inicial

```
[URL do servidor]?page=Commander&refPath=<caminho>
```

#### Cronograma

```
[URL do servidor]?page=Agenda&refPath=<caminho>
```

#### Calendário

```
[URL do servidor]?page=Calendar&refPath=<caminho>
```

#### Ícone personalizado

```
[URL do servidor]?page=CustomIcon&refPath=<caminho>
```

#### Operador personalizado

```
[URL do servidor]?page=Template&refPath=<caminho>
```

#### Conjunto de dados

```
[URL do servidor]?page=Dataset&refPath=<caminho>
```

#### Pacote

```
[URL do servidor]?page=C20Package&refPath=<caminho>
```

### Exibição de processos

[URL do servidor]?page=AppMonitor&refPath=<caminho>

### Recurso

[URL do servidor]?page=Resources&refPath=<caminho>

### Processo

[URL do servidor]?page=processeditor&refPath=<caminho>

### Outras entidades:

#### Exibição de processos (padrão)

O seguinte formato acessa a Exibição de processos, conforme visto na guia Operações:

[URL do servidor]?page=processwatch

#### Lista de tarefas

O seguinte formato acessa *todas* as tarefas, conforme visto na guia Operações:

[URL do servidor]?page=tasklist&tasklist=alltasks

O seguinte formato acessa as tarefas do *grupo*, conforme visto na guia Operações:

[URL do servidor]?page=tasklist&tasklist=grouptasks

O seguinte formato acessa *minhas* tarefas, conforme visto na guia Operações ou na guia Início:

[URL do servidor]?page=tasklist

#### Instâncias de processo

O seguinte formato acessa uma instância em tempo de execução de um processo em sua própria janela:

[URL do  
servidor]?page=runtimeeditor&ROID=<ID\_da\_instância\_em\_tempo  
\_de\_execução>

#### Formulários

O seguinte formato acessa o formulário de solicitação inicial ou o formulário de solicitação de interação usado para responder a uma tarefa:

[URL do  
servidor]?ROID=<runtime\_object\_ID>&tasklist=ALL\_TASK\_FILTER  
&page=replytask

O seguinte formato acessa uma lista de instâncias de formulário de solicitação inicial, conforme visto na guia Operações:

[URL do servidor]?refPath=<path\_to\_SRF\_object>&page=srflist

### Versões de objeto

Para acessar uma versão disponibilizada específica de qualquer objeto da biblioteca, com exceção de um processo, acrescente o parâmetro *versionid* ao URL:

```
&versionid=<version_number>
```

Por exemplo, para acessar a versão 3 do objeto MyResource:

```
[URL do  
servidor]?page=Resources&refPath=/TestFolder/MyResource&ver  
sionid=3
```

### Versões de processos

Para acessar uma versão disponibilizada específica de um processo, acrescente o parâmetro *version* ao URL:

```
&version=<version_number>
```

Por exemplo, para acessar a versão 4 do objeto MyProcess:

```
[URL do  
servidor]?page=processeditor&refPath=/TestFolder/MyProcess&  
version=4
```

## Navegar até uma parte específica de um processo

Ao trabalhar com processos longos, use a janela Navegador para ajudá-lo a ajustar a exibição atual. Aplicar uma panorâmica a uma exibição menor de um processo é mais conveniente do que rolar pelo processo inteiro na janela principal.

### Siga estas etapas:

1. Abra um processo.
2. No Criador de processos, selecione Navegador no menu Exibir.  
A janela Navegador mostra uma imagem em miniatura do seu processo.
3. Arraste o quadro retangular até sobre a parte do processo que você deseja exibir.  
A janela principal mostra uma exibição ampliada da área selecionada.
4. Para ajustar a exibição em miniatura do seu processo, arraste o quadrado no canto inferior direito.
5. Para redimensionar a paleta Navegação, arraste qualquer borda ou canto do mesmo modo como você redimensiona qualquer paleta.

## Multilocação e CA Process Automation

Em uma implantação de multilocação, os administradores desejam controlar o acesso do usuário às instâncias de processo com base em um inquilino ou em um conjunto de inquilinos. É possível usar processos comuns do CA Process Automation em vários inquilinos. Esse recurso permite controlar o acesso das instâncias do processo. É possível impedir que um usuário com acesso a um locatário acesse uma instância relacionada a um inquilino diferente. Esse recurso também resulta em uma duplicação e sincronização do Processo avançado e reduz as tarefas de manutenção relacionadas.

É possível limitar o acesso às instâncias de processo, com base no acesso a um inquilino, usando processos comuns nos inquilinos. Isso é realizado definindo uma ID de inquilino (como uma nova variável) como parte de uma instância de processo. Em seguida, valide o acesso a essa ID de inquilino quando o acesso a essa instância de processo for solicitado.

**Observação:** a configuração da ID de inquilino é responsabilidade do criador de processos como parte da criação do processo ou como parte dos parâmetros de entrada para o processo. Qualquer processo que não tenha uma ID de inquilino já definida deverá seguir a restrição atual de controle de acesso com base na definição do processo.

## Criar um processo existente ciente dos multilocação

Tornar um processo existente ciente dos vários inquilinos envolve as seguintes etapas:

- Adicionar uma variável relacionada à segurança ao contexto de uma instância de processo em execução em tempo de execução
- Criar uma diretiva para multilocação

### Siga estas etapas:

1. Para adicionar uma variável relacionada à segurança ao contexto de uma instância de processo em execução em tempo de execução, execute uma das seguintes ações:

- Adicione uma variável relacionada à segurança ao contexto de uma instância de processo em execução em tempo de execução usando JavaScript:

```
Process.SECURITY_CONTEXT_ID=<ID>;
Process.SECURITY_CONTEXT_GRP=[set the product group or family];
```

Por exemplo:

```
Process.SECURITY_CONTEXT_ID="myid";
Process.SECURITY_CONTEXT_GRP="mygrp";
```

- Transmita valores para a instância do processo usando tags <params> ao iniciar o processo por meio do serviço web executeProcess. Por exemplo:

```
<executeProcess xmlns="http://www.ca.com/itpam">
  <flow>
    ...
  <params>
    <param name="SECURITY_CONTEXT_ID">myid</param>
    <param name="SECURITY_CONTEXT_GRP">mygrp</param>
  </params>
</executeProcess>
```

2. Para criar uma diretiva para multilocação, execute as seguintes etapas:
  - a. Adicione uma diretiva no EEM usando Objeto como o nome da classe de recursos.
  - b. Especifique as identidades para as quais esta diretiva é válida.
  - c. Especifique Recursos para que essa diretiva corresponda ao caminho do processo para o qual você está criando a diretiva.
  - d. Especifique as ações de diretivas necessárias.
  - e. Adicione filtros e especifique valores para os atributos nomeados ID de contexto de segurança e Grupo de contexto de segurança.

Lógica	(	Tipo/valor à esquerda	Operador	Tipo/valor à direita
NENHUM		atributo nomeado Security Context ID	STRING EQUAL ==	valor myid
E		atributo nomeado Security Context ID	STRING EQUAL ==	valor mygrp

Adicionar filtro

Essa diretiva aplica-se a todas as instâncias de processo quando os valores dos atributos nomeados corresponderem aos da instância do processo em tempo de execução.

## Herdar segurança em subprocessos

Os subprocessos herdam automaticamente os valores dos atributos de segurança (ou seja, SECURITY\_CONTEXT\_ID e SECURITY\_CONTEXT\_GRP) de seu processo pai.

## Adicionar variáveis na hora da inicialização

É possível especificar novos valores na opção Código de inicialização do conjunto de dados do processo do operador de serviço Executar processo. Os valores para o contexto de segurança especificado no Código de inicialização do conjunto de dados do processo têm precedência sobre os especificados no processo pai.

## Processos de multilocação usando a exibição de processos

É possível exibir as instâncias de processo em tempo de execução na Exibição de processos se uma diretiva de multilocação permitir. Também é possível executar ações necessárias, como Abrir, Cancelar, Suspende, e assim por diante, se uma diretiva permitir.

## O editor de código do CA Process Automation

O editor de código avançado no CA Process Automation permite que você facilmente crie, edite e depure código de várias linguagens de scripts nos seguintes locais:

- O código de execução anterior e posterior de um operador.
- O parâmetro de entrada do código-fonte para o operador Executar o script.

- O parâmetro de entrada do código-fonte para o operador Executar o JavaScript.
- O parâmetro de código de inicialização de dados para o operador Iniciar processo.
- Editores de script de formulário para o formulário de solicitação de interação e o formulário de solicitação inicial.
- O parâmetro de entrada do código-fonte para o operador Método SOAP.
- O parâmetro de entrada do código-fonte para o operador Solicitar método SOAP.
- O parâmetro de entrada do código-fonte para o operador Aplicar XSLT.
- O parâmetro de entrada do código-fonte para qualquer operador personalizado em que o operador de base seja um dos operadores mencionados anteriormente.

O editor de código oferece suporte aos seguintes idiomas:

- JavaScript (.js) (disponível para JavaScript do lado do cliente e JavaScript do lado do servidor)
- XML (.xml)
- SQL (.sql)
- Java (.java)
- batch (.bat)
- Windows Command File (.cmd)
- Visual Basic Script (.vbs)
- Windows Script Host Settings (.wsh)
- Arquivo PerlScript (.ps)
- HTML (.html)
- Shell (.sh)

Alguns operadores adicionais contêm editores de texto que usam a funcionalidade de editor de código, quando aplicável. Quando o editor de código e o CA Process Automation oferecem suporte à linguagem específica usada no operador, os editores de texto automaticamente se adaptam à linguagem determinada. Especificamente, os editores de texto realçam palavras-chave específicas da linguagem. No entanto, se os operadores usam uma linguagem que *não* tem suporte no editor de código ou que não foi implementada pelo CA Process Automation, um editor de texto simples será disponibilizado para a codificação.

Se um processo estiver disponibilizado, o editor de código é iniciado no modo somente leitura e o editor e a barra de ferramentas são desativados. No entanto, alguns botões da barra de ferramentas (por exemplo, Pesquisar, Ajuda e Validação de script) ainda estão disponíveis.

O editor de código inclui os seguintes recursos:

**Arrastar e soltar**

Arraste qualquer arquivo ou texto selecionado (de dentro ou de fora do CA Process Automation) e solte-o no editor para preenchê-lo.

**Search/Replace**

Localizar um código específico (com as opções Diferencia maiúsculas de minúsculas e Expressões regulares) e, em seguida, substituir instâncias específicas ou todas as ocorrências.

**Desfazer**

Reverter a ação anterior.

**Refazer**

Reverter a ação anterior Desfazer.

### **Ir para a linha número**

Avançar para uma determinada linha de código.

### **Reformatar seleção**

Ajustar recuo do texto selecionado.

### **Reformatar o documento inteiro**

Ajustar recuo do documento de código inteiro.

### **Importar a partir do arquivo**

Procurar para localizar um arquivo específico para importar para o editor.

Os modos do editor com suporte para arquivos importados incluem:

- .bat
- .cmd
- .js
- vbs
- .wsh
- .ps1
- .sh
- .pl
- .xml

O CA Process Automation fornecerá um erro se qualquer outra extensão de arquivo for importada no editor de código avançado.

**Observação:** qualquer tipo de arquivo pode ser importado para o editor genérico.

### **Exportar para arquivo**

Salvar o arquivo em seus downloads do CA Process Automation.

### **Mostrar log de erros**

Exibir erros no código. Esse recurso está disponível apenas nos seguintes casos:

- O código de execução anterior e posterior de um operador.
- O parâmetro de entrada do código-fonte para o operador Executar o JavaScript.
- O parâmetro de código de inicialização de dados para o operador Iniciar processo.

O editor destaca o código incorreto em um log de erros de rolagem na parte inferior do editor. Cada erro contém o número da linha e um hiperlink para ir até a linha específica do código. Quando o erro é corrigido, o editor o limpa automaticamente do log de erros.

**Observação:** o código deve ser inserido corretamente, em expressões válidas. No entanto, é possível salvar o código mesmo que ele contenha erros, embora o CA Process Automation emita um aviso. Algumas validações (como os exemplos a seguir) são ignoradas:

**"strict"**

Espera 'Use Strict' na declaração da função JavaScript.

**"eqeqeq"**

Espera '===' em vez de '==' na condição.

**"smarttabs"**

Valida uma combinação de guias e espaços.

**Dobra de código**

Ao clicar no número da linha de qualquer "{", o editor de código dobra o código até o colchete correspondente "}".

Ao editar o XML, a dobra de código baseia-se nas tags de correspondência. Você pode expandir ou recolher o fragmento XML com base nas tags de correspondência. O editor de código fecha automaticamente as tags finais para você durante a edição.

O editor de código também usa as seguintes convenções:

- Exibe os números de linha.
- Remove as linhas vazias.
- Destaca o texto correspondente no documento quando se seleciona um texto específico.
- Destaca a linha atual.

## Código de cores

O editor de código usa as seguintes cores para exibir o JavaScript:

### Laranja

Indica:

- Palavras-chave de conjunto de dados para o JavaScript do lado do servidor
- Variáveis de formulários para o JavaScript do lado do cliente

Exemplo: **Datasets**

### Azul

Indica as funções do sistema do CA Process Automation. O JavaScript do lado do cliente e do lado do servidor possui diferentes conjuntos de funções do sistema.

Exemplo: `adjustResourceVals`

### Púrpura

Indica as palavras-chave do JavaScript.

Exemplo: `comment`

## Atalhos

O editor de código usa os seguintes atalhos:

- Ctrl+C (cópia)
- Ctrl+V (colar)
- Ctrl+X (recortar)
- Ctrl+Z (desfazer)
- Ctrl+Y (refazer)
- Ctrl+A (Selecionar tudo)
- Excluir (para excluir texto selecionado)
- Início (para ir para o início da linha atual)
- Término (para ir para o término da linha atual)

Os seguintes atalhos são válidos somente para JavaScript:

- Pressione Ctrl+Espaço para exibir os seguintes itens:
  - Uma lista de palavras-chave do conjunto de dados (exibida primeiro) e palavras-chave de JavaScript para o JavaScript do lado do servidor.
  - Uma lista de variáveis de formulários (exibida primeiro) e palavras-chave de JavaScript para o JavaScript do lado do cliente.
- Pressione Ctrl+Alt para exibir uma lista de funções do sistema. O JavaScript do lado do cliente e do lado do servidor possui diferentes conjuntos de funções do sistema.

**Observação:** apenas o JavaScript do cliente pode ser usado no Criador de formulário (para formulários de solicitação inicial e formulários de solicitação de interação). As funções de sistema específicas ao CA Process Automation não estão disponíveis. No entanto, as funções de formulário estão disponíveis.

# Capítulo 5: Operadores e ícones

---

O CA Process Automation executa as instruções nos operadores adicionados a um processo ou uma programação. Esta seção descreve como configurar os operadores depois de tê-los adicionado a um objeto de processo ou programação. Ela também inclui informações sobre conectores e operadores personalizados.

Esta seção contém os seguintes tópicos:

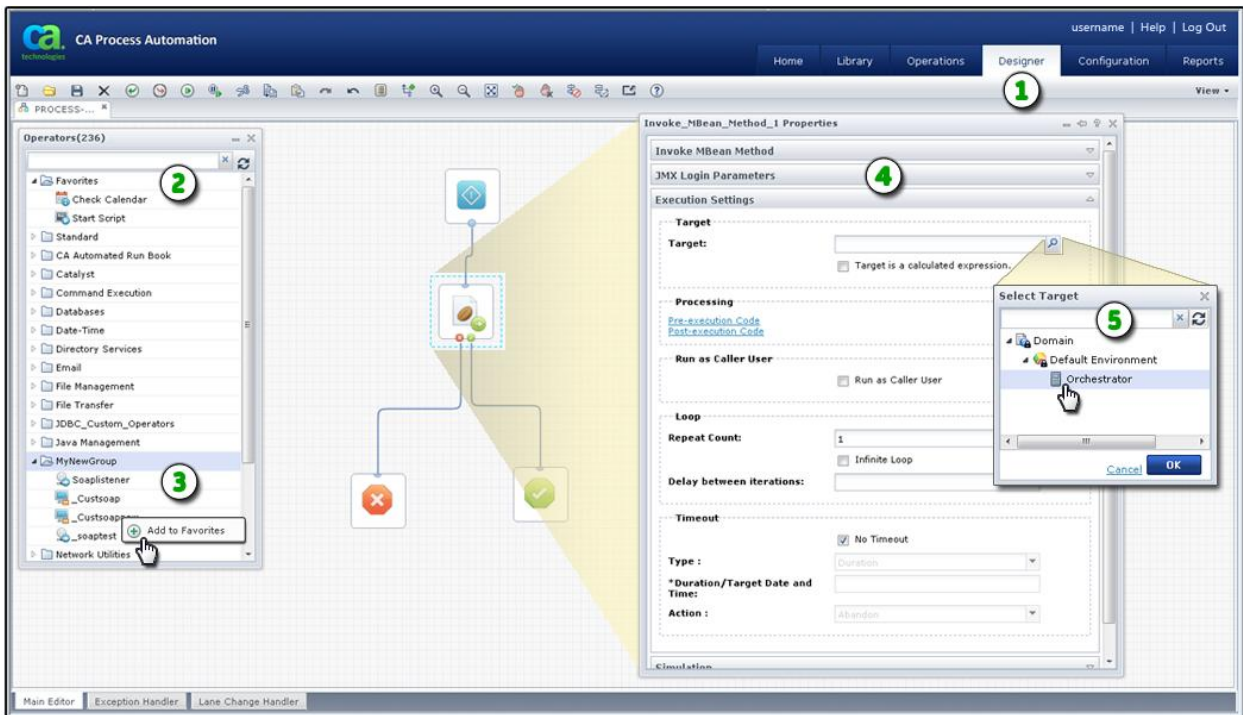
[Operadores](#) (na página 154)

[Ícones dos operadores](#) (na página 192)

## Operadores

Uma seção anterior deste guia apresentou [operadores de processo](#) (na página 104). Para saber mais sobre cada operador, consulte a *Referência do Criador de Conteúdo*.

Esta seção fornece diretrizes gerais para configurar propriedades e trabalhar com tipos diferentes de operadores, incluindo operadores personalizados.



**Item:**    **Descrição:**

1

**Barra de ferramentas e guia do criador:** quando você abrir um novo processo ou editar um processo existente no Navegador da biblioteca, a guia Criador será exibida. Na barra de ferramentas, clique em Exibir e selecione as paletas Operadores e Propriedades.

2

**Paleta Operadores:** procure operadores por pasta ou nome.

3

**Operadores personalizados em um grupo personalizado:** arraste e solte um operador personalizado dessa paleta para o layout do processo. Clique com o botão direito do mouse em um operador para adicioná-lo ou removê-lo no grupo Favoritos.

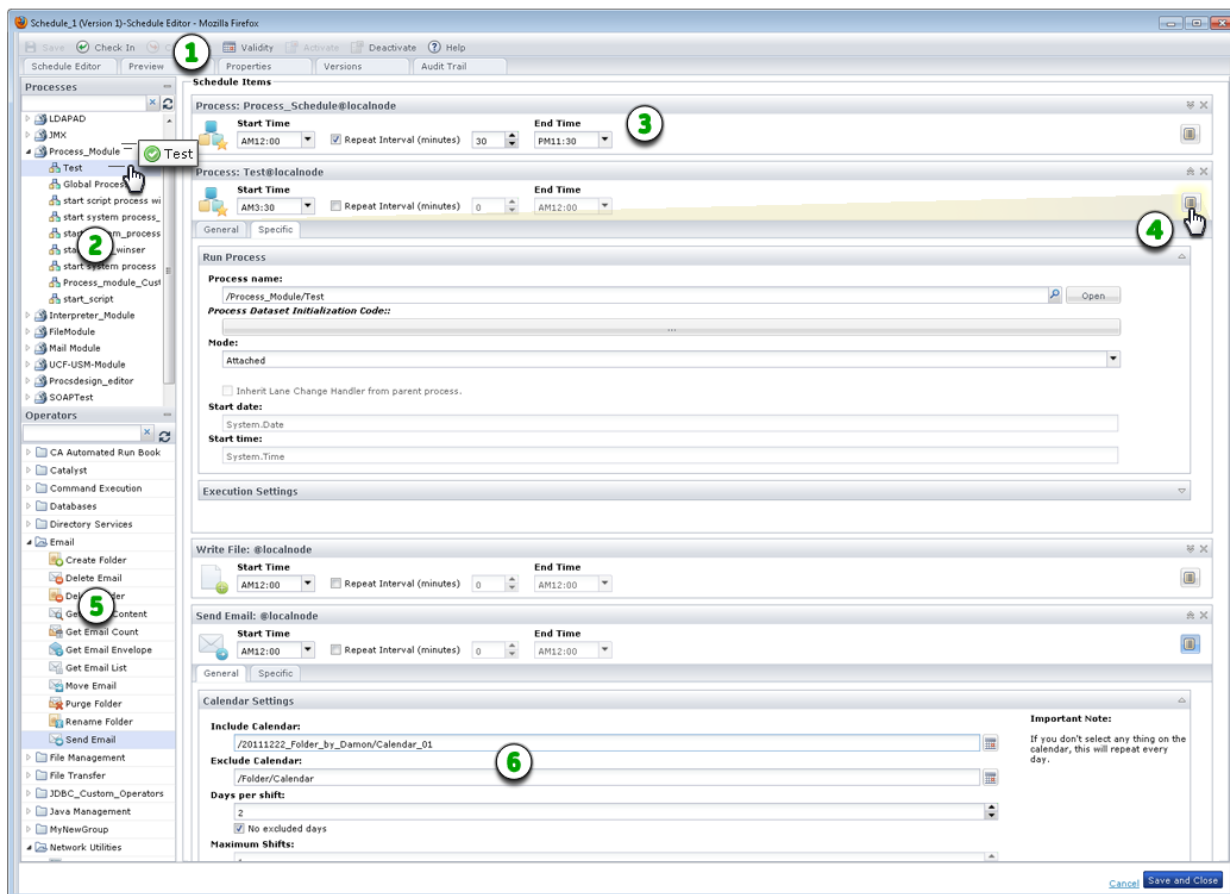
4

**Paleta Propriedades:** use esta paleta e seus botões, links e janelas para gerenciar as propriedades do operador selecionado. Expanda e recolha painéis à medida que trabalha. Cada painel consiste em propriedades semelhantes.

5

**Janelas adicionais:** algumas propriedades são exibidas em suas próprias janelas.

Você também trabalha com operadores para configurar programações.



**Item:**    **Descrição:**

- 1 **Editor de programação:** quando você abrir uma programação no Navegador da biblioteca, o Editor de programação é exibido. Use a barra de ferramentas para ativar, definir a validade, disponibilizar ou salvar a programação.
- 2 **Painel de processos:** selecione os processos que deseja incluir e arraste-os para a página Programar itens.
- 3 **Processo programado na fila:** defina a duração e a frequência para executar o processo em um único dia.
- 4 **Propriedades do processo:** clique em Propriedades para exibir as propriedades do processo nas guias Geral e Específico.
- 5 **Painel Operadores:** selecione os operadores que deseja incluir e arraste-os para a página Programar itens.
- 6 **Propriedades do operador:** clique em Propriedades para exibir as propriedades do operador nas guias Geral e Específico.

## Configurar propriedades do operador

As propriedades do operador são exibidas no Criador de processos na paleta Propriedades. Campos semelhantes são organizados em caixas de grupo de painéis expansíveis com títulos familiares, como Processo, Configurações de execução, Simulação e Ícone. Alguns painéis e grupos são comuns a todos os operadores e outros são exclusivos para um determinado operador. Por exemplo, as propriedades para cada operador incluem um painel Ícone. No entanto, o painel Dados de chamada SOAP assíncrona aparece apenas para o operador Chamar método SOAP assíncrono.

As propriedades do operador também estão disponíveis quando você inclui operadores nos objetos Programação.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Abra um objeto de processo ou um objeto de programação.  
Os processos são abertos no Criador de processos. As programações são abertas na caixa de diálogo Editor de programação.
3. Na barra de ferramentas, clique em Reservar se o objeto ainda não foi reservado.
4. Adicione um operador ao processo ou à programação:
  - a. Em um processo, arraste um operador para o layout do processo da paleta Operadores.
  - b. Em uma programação, recolha o painel Processos, expanda o painel Operadores e arraste um operador para a lista Programar itens.
5. Exiba as propriedades do operador:
  - a. Em um processo, clique duas vezes no operador.  
A paleta Propriedades é exibida.
  - b. Em uma programação, clique em Propriedades e nas guias Geral ou Específico.
6. Clique em Salvar.  
Os valores de propriedades que você digitou para o operador selecionado são salvos.

**Observação:** para obter informações sobre as propriedades do operador, consulte o *Guia de Referência do Criador de Conteúdo*.

## Recuperação automática

O recurso *Recuperação do operador* controla quais processos podem ser recuperados como parte da recuperação automática ou manual de um touchpoint, orquestrador ou grupo de hosts. Você pode ativar ou desativar esse recurso.

- Ative a recuperação em um processo e defina um manipulador de exceções que notifique você sobre uma falha.
- Desative a recuperação em um processo e defina um manipulador de exceção que corrija o erro de sistema. Por exemplo, o manipulador poderia executar o operador em outro touchpoint.

**Observação:** para obter mais informações, consulte [Manipulação de exceções](#) (na página 431).

O CA Process Automation verifica as *configurações de Ativar recuperação do operador* para cada objeto antes de executar uma instância de processo. O CA Process Automation usa as configurações para determinar se a instância de processo pode ser recuperada. A instância de processo não será afetada se as configurações de objeto de biblioteca mudarem após a execução da instância de processo.

Se ativado, o procedimento de recuperação agirá sobre os operadores e recuperará as instâncias de processo que falharam com `SYSTEM_ERROR`. Os processos dos operadores devem ser definidos para serem recuperados e devem estar no estado *Bloqueado*, *Em execução* ou *Aguardando* quando a recuperação for disparada. A recuperação do operador redefine o operador e, em seguida, retoma os processos. Um operador no estado *Bloqueado* deve retomar a operação e ser reexecutado durante a recuperação do touchpoint.

Os novos processos criados no CA Process Automation versão 4.0 ou posterior têm essa opção selecionada por padrão. A opção *Ativar recuperação do operador* está desmarcada, por padrão, apenas para os processos existentes criados antes de uma atualização do CA Process Automation versão 4.0.

## Java e JARs externos

O módulo Java permite especificar o código Java a ser executado dentro de um intérprete *BeanShell* na JVM do CA Process Automation. O BeanShell é um intérprete de origem Java dinâmico incorporado, linguagem de script e ambiente flexível. Usando os operadores e o módulo Java do CA Process Automation, você pode:

- Importar e fazer referência a arquivos JAR no nível de módulo ou de operador.
- Defina configurações globais para todos os operadores Executar código Java ou configurações específicas para um determinado operador. Você poderá especificar os caminhos para os arquivos JAR externos que os operadores podem usar. Também poderá definir a configuração de log padrão no módulo. Para cada operador, você pode especificar o código que deseja executar, os parâmetros de entrada e os nomes das variáveis de saída. As configurações de log especificadas para um único operador Executar código Java substituem as configurações de log no nível de módulo.
- Escreva o código Java que faz referência às classes nesses JARs. O CA Process Automation cria automaticamente novos tipos de dados Objeto Java durante a execução de um programa Java. Como novos objetos Java são chamados dentro do operador Executar código Java, você não precisa usar nenhum método de criação ou destruição de objeto. Após a execução do operador, o CA Process Automation coleta automaticamente qualquer recurso, classe e objeto na memória que estiver na lixeira.
- Execute o código Java usando o operador Executar código Java. Você pode chamar as classes de um arquivo JAR (Java Archive) externo a partir de um operador Executar código Java. Use esse operador para aproveitar a funcionalidade oferecida pelo seu código Java existente.
- Salve objetos Java no conjunto de dados do operador para disponibilizá-los aos operadores subsequentes Executar código Java. Você pode optar por salvar um objeto Java inteiro no conjunto de dados do operador antes do término da execução do operador. Em seguida, você poderá utilizar o objeto Java salvo no conjunto de dados do operador em operadores subsequentes, passando-o para eles.

## Configurar o módulo Java

Você pode chamar as classes de um arquivo JAR (Java Archive) externo a partir de todos os operadores Executar código Java. Configure o módulo Java para aplicar configurações a esses operadores e aproveitar a funcionalidade oferecida pelo seu código Java existente. Em seguida, use o operador Executar código Java para criar um objeto Java.

### Siga estas etapas:

1. Especifique os JARs com os quais deseja trabalhar em uma máquina com orquestrador ou agente do CA Process Automation. Localize o campo Jars externo padrão para o módulo. Digite os caminhos para os JARs externos a serem carregados para uso por todos os operadores Executar código Java em execução no orquestrador ou no agente. Para cada caminho, você pode digitar:

- a. O caminho completo de um arquivo JAR residente na máquina em que o orquestrador ou o agente está sendo executado. O caminho completo é iniciado com uma das seguintes barras:

/  
\\

Você também pode designar o caminho completo usando uma expressão regular que inicie com um caractere seguido por dois-pontos (:) e, em seguida, o restante da sequência de caracteres, incluindo a sintaxe de ponto, como em:

^.:.\*

- b. O caminho completo de um arquivo JAR disponível em `http://` ou `https://`. O caminho não exige autenticação e não deve ser acessível por meio de um proxy `http`.
- c. Um caminho relativo de um arquivo JAR que foi carregado para a pasta Recursos do usuário do CA Process Automation. A menos que você especifique um caminho completo, o aplicativo considerará o caminho inserido como um caminho relativo.

O Carregador de classe do Módulo Java, que todos os operadores *Executar código Java* em execução do módulo Java compartilham, carrega os JARs que você lista nesse campo. Os JARs são carregados uma vez quando estiverem executando o primeiro operador de módulo Java e, depois, após qualquer alteração feita nos JARs externos padrão. Qualquer JAR inserido nessa lista ficará disponível ao código Java executado pelos operadores *Executar código Java*. Todas as classes definidas nos JARs no nível de operador substituirão as mesmas classes especificadas nos JARs no nível de módulo.

2. Além dos JARs externos, digite os caminhos dos arquivos `.class` que todos os operadores Executar código Java em execução no orquestrador ou no agente estejam usando.

- a. Para os arquivos .class em um pacote sem nome, digite um caminho que termine com o diretório que contém os arquivos .class. Por exemplo, MyAccount.java não pertence a um pacote e MyAccount.class está no seguinte local:

C:\java\tests\MyAccount.class

Defina o operador para usar o seguinte caminho:

C:\\java\\tests

- b. Para os arquivos .class em um pacote nomeado, digite um caminho que termine com o diretório que contém o pacote *raiz*. O pacote raiz é o primeiro pacote no nome completo do pacote. Por exemplo, MyAccount.java pertence ao pacote com.ca.tech. MyAccount.class está no seguinte caminho:

C:\java\othertests\com\ca\tech\MyAccount.class

Defina o operador para usar o seguinte caminho:

C:\\java\\othertests

**Observação:** especifique o caminho de uma pasta como um caminho completo ou relativo para Recursos de usuário do CA Process Automation. Não especifique um caminho http. Especifique o caminho de uma pasta para carregar arquivos .class, e não arquivos JAR. Diferentemente dos arquivos .class, cada arquivo JAR requer um caminho separado que termine com o arquivo JAR (e não o diretório onde ele reside).

3. (Opcional) Faça upload dos JARs com os quais deseja trabalhar para Recursos de usuário do CA Process Automation.

O CA Process Automation espelham automaticamente os JARs.

**Observação:** os recursos, incluindo recursos do usuário, são espelhados no intervalo de espelhamento do orquestrador ou do agente. Certifique-se de que os arquivos JAR carregados nos recursos de usuário já estejam espelhados antes de usá-los nos operadores do módulo Java.

4. Verifique se os arquivos JAR contendo as classes Java com as quais deseja trabalhar estão disponíveis para o orquestrador/agente cujo touchpoint está executando o operador.
5. (Opcional) Configure o agente de log padrão do módulo. É possível substituir essa configuração no nível de operador.
6. (Opcional) Substitua qualquer configuração no nível de módulo, definindo operadores individuais. Consulte o próximo procedimento, [Configurar um operador Executar código Java](#) (na página 161).
7. Execute qualquer operador Executar código Java. O módulo Java captura todos os erros ou exceções encontrados durante uma operação e alerta o usuário no campo Motivo do operador Problema.

## Configurar o operador Executar código Java

Você pode chamar as classes de um arquivo JAR (Java Archive) externo a partir de um operador Executar código Java. O operador Executar código Java pode usar JARs no nível de operador e no nível de módulo. Também é possível configurar o operador Executar código Java para substituir as configurações no nível de módulo. Use o operador para aproveitar a funcionalidade no seu código Java existente.

### Siga estas etapas:

1. Se você ainda não tiver configurado o módulo Java, conclua essa tarefa primeiro. Consulte [Configurar o módulo Java](#) (na página 159).

2. Configure o operador Executar código Java.

Especifique os caminhos para os JARs externos usados pelo operador Executar código Java. Para cada caminho, você pode digitar:

- O caminho completo para um arquivo JAR residente no host em que o orquestrador ou o agente está sendo executado. O caminho completo é iniciado com uma das seguintes barras:

/  
\\

Você também pode usar uma expressão regular (incluindo notação de ponto) que inicia com um caractere, dois-pontos (:) e, em seguida, a cadeia de caracteres restantes, como em:

^.:.\*

- O caminho completo de um arquivo JAR disponível em `http://` ou `https://`. O caminho não pode exigir autenticação e não deve ser acessível por meio de um proxy http.
- Um caminho relativo para um arquivo JAR que reside na pasta Recursos do usuário do CA Process Automation.

A menos que você especifique um caminho completo, o aplicativo considerará o caminho inserido como um caminho relativo.

O operador carrega os JARs listados nesse campo e torná-os disponíveis ao código Java no operador em execução. As classes definidas nesses JARs substituem as mesmas classes especificadas nos JARs no nível de módulo.

3. Além dos JARs externos, digite os caminhos para qualquer arquivo `.class` do operador Executar código Java.
  - a. Para os arquivos `.class` em um pacote sem nome, digite um caminho que termine com o diretório que contém os arquivos `.class`. Por exemplo, se `MyAccount.java` não pertencer a um pacote e `MyAccount.class` estiver em:

`C:\java\tests\MyAccount.class`

Defina o operador para que ele use o seguinte caminho:

C:\\java\\tests

- b. Para os arquivos .class em um pacote nomeado, digite um caminho que termine com o diretório que contém o pacote raiz. O pacote raiz é o primeiro pacote no nome completo do pacote. Por exemplo, se MyAccount.java pertencer ao pacote com.ca.tech e MyAccount.class estiver em:

C:\\java\\othertests\\com\\ca\\tech\\MyAccount.class

Defina o operador para que ele use o seguinte caminho:

C:\\java\\othertests

**Observação:** especifique o caminho de uma pasta como um caminho completo ou relativo para Recursos de usuário do CA Process Automation. Não especifique um caminho http. Especifique o caminho para uma pasta para que ela carregue arquivos .class, não arquivos JAR. Ao contrário dos arquivos .class, cada arquivo JAR requer um caminho separado que termina com o arquivo JAR (não o diretório onde ele reside).

4. (Opcional) Faça upload dos JARs com os quais deseja trabalhar para Recursos de usuário do CA Process Automation.

O CA Process Automation espelham automaticamente os JARs.

**Observação:** os recursos, incluindo recursos do usuário, são espelhados durante o intervalo de espelhamento do orquestrador ou do agente. Certifique-se de que os arquivos JAR carregados nos recursos de usuário já estejam espelhados antes de usá-los nos operadores do módulo Java.

5. Especifique o código a ser executado.
6. Especifique os parâmetros de entrada a serem passados ao código Java.
7. Especifique os nomes de saída das variáveis criadas no código Java do operador. Os nomes de variáveis de saída devem ser salvos no conjunto de dados do operador quando a execução do código é concluída.

**Observação:** o CA Process Automation serializa objetos Java que não sejam booleanos, data, número inteiro, número, sequência de caracteres ou uma matriz desses tipos e os salva como JavaObjects.

8. (Opcional) Especifique a configuração do agente de log desse operador Executar código Java. Essas configurações substituem as configurações do agente de log no nível de módulo.
9. Execute o operador Executar código Java.

O módulo Java captura exceções ou erros encontrados durante uma operação e alerta o usuário no campo Motivo do operador de problema.

## Usando um `JsonObject`

Os objetos Java são salvos após um operador Executar código Java ser concluído em um tipo de dados `JsonObject`. Você pode usar uma variável de conjunto de dados `JsonObject` das seguintes maneiras:

- Passar a variável do conjunto de dados `JsonObject` na lista de parâmetros do operador Executar código Java.
- Passar o caminho da variável do conjunto de dados `JsonObject` em uma variável de sequência de caracteres a partir de um Formulário de solicitação de interação ou de um Formulário de solicitação inicial. Por exemplo:  
`JsonObjectPath = Datasets["GlobalDatasets"].acct`  
Em seguida, você pode usar a função `eval` ao passar a variável `JsonObjectPath` para a lista de parâmetros do operador Executar código Java. Por exemplo:  
`eval(Process.IRF.JsonObjectPath)`
- Copie um `JsonObject` no Javascript. Você também pode atribuir um `JsonObject` para outro.

**Observação:** não modifique o valor real de um `JsonObject` do CA Process Automation (a sequência de caracteres serializada do objeto Java original). As suas alterações podem impedir que o operador Executar código Java carregue o `JsonObject`.

Observe as seguintes restrições ao trabalhar no JavaScript:

- Assim como em senhas, você não pode concatenar um `JsonObject` com uma sequência de caracteres.
- Não é possível carregar um `JsonObject` e usar seus métodos no JavaScript. Em vez disso, passe o `JsonObject` na lista de parâmetros do operador Executar código Java e acesse-o no código Java do operador.

O código Java escrito por você pode consistir em expressões e instruções Java normais. Você também pode definir seus próprios métodos e usá-los no código. Por exemplo:

```
// Importe as classes que deseja usar
import ca.tech.pam.MyAccount;
// Observação: não é necessário importar StringBuffer e Date, pois eles fazem parte
dos
// pacotes importados automaticamente
// import java.lang.StringBuffer;
// import java.util.Date;
// Observação: o jar que contém a classe ca.tech.pam.MyAccount
// deve estar na lista de Jars externos do operador ou do módulo;
// mas java lang e java util estão em rt.jar, que é automaticamente colocado no
caminho de classe

MyAccount acct = new MyAccount(1000,00);

// Use os métodos públicos do objeto MyAccount
acct.addFunds(34.44);
acct.subFunds(10);

// Defina seu próprio método
String getStatement(MyAccount acc) {
    StringBuffer strBuff = new StringBuffer("Account Balance: " +
acc.getBalance());
    Date dt = new Date(System.currentTimeMillis());
    strBuff.append(" on date: " + dt);
    return strBuff.toString();
}
// Use o método que você definiu
// imprima também a instrução usando o objeto 'agente de log '
// configurado na página 'Agente de log' do operador
logger.info(getStatement(acct));
```

Depois de executar esse código Java, a mensagem de log mostra o saldo da conta, a data e a hora:

```
Account Balance: 124.44 on date: Wed Jul 13 12:53:37 EDT 2011
```

## Operadores personalizados

Você pode definir objetos *operador personalizado* que os usuários podem adicionar a seus processos ou às suas programações da mesma forma que adicionariam qualquer outro operador. Você cria operadores personalizados reutilizando um operador de base existente. Defina as configurações desse operador de base na paleta Propriedades. Use operadores personalizados para compartilhar a nova funcionalidade em toda a organização de TI.

Por exemplo, você pode usar o operador Obter variável de SNMP como base para um operador personalizado que recupera informações específicas de um roteador de rede. Configure a OID adequada da variável de SNMP e a sequência de caracteres de comunidade e, em seguida, defina as propriedades de modo que um usuário possa configurar o endereço IP da variável de SNMP. Depois que ela for publicada, os usuários poderão adicionar o operador personalizado a um processo ou programação e, em seguida, configurar o endereço IP para acessar informações de roteamento de rede específicas. Os usuários não precisam conhecer a OID da variável de SNMP ou a sequência de caracteres de comunidade. Esses valores já foram configurados e estão provavelmente ocultos do usuário.

Um uso mais eficiente dos operadores personalizados está no desenvolvimento de interfaces para novos sistemas e aplicativos corporativos, incluindo:

- Aplicativos críticos à missão; as organizações frequentemente desenvolvem e implantam esses aplicativos internamente.
- Serviços web (SOAP e RESTful)
- Aplicativos de linha de comando e script.
- Bancos de dados
- Java

Desenvolva operadores personalizados que executem ações comuns que façam interface com seus aplicativos e sistemas. Reutilize seus próprios operadores personalizados a fim de permitir que sua organização trabalhe facilmente com esses aplicativos e sistemas de processos automatizados. Os usuários podem executar ações sem o conhecimento especializado de todas as interfaces do aplicativo e do sistema.

Os operadores personalizados também fornecem a vantagem de se adaptarem facilmente às alterações do ambiente de TI. Por exemplo, você precisa alterar apenas o objeto operador personalizado original se as alterações no ambiente de TI exigirem que você altere a sua interface com um aplicativo. Todos os processos e as programações que usam o operador personalizado aplicam automaticamente as alterações na versão designada mais recente. Dessa forma, as alterações no ambiente de TI são transparentes para todos os processos e programações.

## Criar um objeto Operador personalizado

É possível criar operadores personalizados em qualquer pasta de biblioteca.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique em uma pasta.
3. Na barra de ferramentas, clique em Novo e escolha Operador personalizado.  
A caixa de diálogo *Selecionar o operador de base* é exibida.
4. Selecione o operador base para seu operador personalizado e clique em OK.  
Expand a pastas ou digite um nome de operador para pesquisa.  
Um objeto Operador personalizado com um nome padrão é criado na biblioteca.
5. Renomeie o operador personalizado.

**Observação:** o aplicativo identifica operadores personalizados por um local exclusivo na biblioteca usando o nome e o caminho do objeto. Evite alterar o local e o nome do objeto do operador personalizado se ele estiver sendo usado em um processo. Renomear ou mover um operador personalizado enquanto ele estiver sendo usado em um processo pode resultar na perda de alterações ou atualizações.

## Propriedades do operador personalizado

Você pode configurar um operador personalizado modificando as configurações padrão disponíveis para o operador de base. Também é possível adicionar parâmetros e páginas de entrada para apresentar configurações de entrada específicas a um usuário do operador personalizado.

Há sete tipos principais de propriedades para cada operador personalizado:

- Formulário
- Visualizar
- Configurações
- Conjunto de dados
- Configuração do grupo
- Propriedades
- Versões
- Trilha de auditoria

Para cada configuração, você normalmente:

- Deixa-a em branco e permite que o usuário configure-a quando ele usar seu operador personalizado.
- Configura-a com um valor predefinido e a marca como invisível, o que oculta toda a configuração do usuário quando este usa o operador personalizado.

Também é possível:

- Configurá-la com um valor predefinido, mas permitir que o usuário altere o valor.
- Configurá-la com um valor predefinido e marcá-la como somente leitura, o que permite que o usuário veja o valor, mas não o edite.

#### **Exemplo: um operador personalizado básico**

É possível criar um operador personalizado que recupere uma informação específica de um roteador de rede usando o operador Obter variável de SNMP como base. Como parte de seu operador personalizado, configure a OID adequada da variável de SNMP e a sequência de caracteres de comunidade e, em seguida, defina as propriedades, de modo que um usuário possa configurar o endereço IP da variável de SNMP. O operador Obter variável de SNMP tem as seguintes configurações específicas:

- Host do agente (Endereço IP)
- Comunidade
- ID do objeto (OID)
- Contagem de tentativas
- Tempo de expiração
- Versão do SNMP

Normalmente, configura-se os valores predefinidos e marca-se todas as configurações como invisíveis, com exceção do Host do agente. Outro criador usando esse operador personalizado vê apenas a configuração Host do agente na paleta Propriedades. Todas as outras configurações estão ocultas. Outros usuários podem especificar o Host do agente para determinar o roteador de rede onde as informações se originam. No entanto, outros usuários não precisam conhecer as outras configurações de um operador Obter variável de SNMP. Contudo que eles configurem o endereço IP correto, as informações em questão são exibidas.

### Exemplo: um operador personalizado avançado

Como um exemplo mais avançado, é possível criar uma interface para um aplicativo interno com o uso de scripts. Neste exemplo, use o operador Iniciar script como base para o operador personalizado. Em seguida, especifique a extensão do script, o script e as outras configurações como os parâmetros a serem passados ao script. Normalmente, esses parâmetros seriam definidos (na verdade, todos os parâmetros que vêm do operador de base) como invisíveis. Também é possível criar sua própria página de configurações para solicitar a um usuário do seu operador personalizado algumas configurações que são específicas para o operador. Como no exemplo simples acima, um usuário final pode usar o operador personalizado para atuar em seu aplicativo interno. Essa técnica estende processos integrados, sem a necessidade de conhecer detalhes específicos sobre como fazer interface com o aplicativo.

#### Mais informações:

[Operador personalizado: guia Propriedades](#) (na página 185)

## Operador personalizado: guia Formulário

Ao criar um operador personalizado, você pode permitir que as páginas e os parâmetros personalizados recebam entradas de usuários finais à medida que eles configuram seu operador personalizado em um processo ou uma programação. Use a guia Formulário do Operador personalizado para:

- Adicionar, remover e renomear páginas de propriedades
- Adicionar, remover e renomear parâmetros nas páginas de propriedades
- Configurar as características de um parâmetro
- Reordenar parâmetros em uma página de propriedades
- Mover parâmetros entre as páginas

Cada operador personalizado pode ter uma ou mais páginas de parâmetros que se baseiam em seu ancestral ou operador de base. Você pode modificar e configurar esses parâmetros. Por exemplo, se a base for o operador Executar o script, você poderá configurar os seguintes parâmetros:

- O campo *scriptType* para definir a extensão do script.
- O campo *inLineScript* para definir o próprio script.
- Os outros campos e parâmetros padrão para o operador Executar o script.

## Adicionar páginas de propriedades

A guia Configurações do operador personalizado inclui as propriedades do operador de base padrão. Estas configurações correspondem aos parâmetros encontrados na paleta Propriedades do Criador de processos. Use a paleta Propriedades para configurar as propriedades de base do operador personalizado e as propriedades personalizadas que você deseja atribuir a ele.

Um operador personalizado pode exigir parâmetros adicionais como entrada na função do operador. É possível adicionar páginas de propriedades para agrupar esses parâmetros adicionais. Ao adicionar páginas para o operador personalizado na guia Formulário, elas são exibidas como seções expansíveis na paleta Propriedades do Criador de processos.

### Siga estas etapas:

1. No Navegador da biblioteca, clique duas vezes em um operador personalizado.
2. Na barra de ferramentas, clique em Reservar.
3. Na janela do criador de operadores personalizados, clique na guia Formulário.
4. No painel Elementos do formulário, expanda todos os seguintes:
  - a. Elementos do formulário
  - b. Layout da página em Elementos do formulário
  - c. Layout da página sob seu operador personalizado
5. Arraste um elemento Página a partir do Layout da página de Elementos do formulário para o nível do Layout da página de seu operador personalizado.
6. Clique na nova página e, em seguida, clique em Renomear na barra de ferramentas. Renomeie a nova página para um identificador significativo.

Cada nome de página corresponde a uma seção nomeada expansível na paleta Propriedades do operador personalizado no Criador de processos.
7. A ordem das páginas na paleta Propriedades também corresponde à ordem exibida na guia Formulário. Selecione uma página e clique em Mover para cima ou Mover para baixo na barra de ferramentas para mover a página para cima ou para baixo na lista.

### Para remover uma página de propriedades, incluindo todos os parâmetros:

1. Selecione a página.
2. Na barra de ferramentas, clique em Excluir.

## Adicionar parâmetros personalizados

Após criar páginas de propriedades, você poderá adicionar parâmetros (campos) a elas. Os parâmetros personalizados são geralmente usados para exibir uma outra apresentação aos usuários para um parâmetro necessário para o operador base. Por exemplo, é possível adicionar um parâmetro para apresentar uma lista de valores que você deseja que os usuários selecionem, em vez de usar uma caixa de edição sem valores definidos. Os parâmetros também são usados para capturar a entrada que será usada em uma expressão para calcular um ou mais parâmetros de operador base. Além disso, os parâmetros são usados para apresentar uma interface aos usuários com a terminologia apropriada.

A menos que você defina suas propriedades Somente leitura, Desativado ou Oculto como verdadeiro, os parâmetros personalizados adicionados ao operador personalizado serão visíveis e configuráveis pelos usuários finais na paleta Propriedades do operador personalizado no Criador de processos ou no Editor de programações.

### Siga estas etapas:

1. No Navegador da biblioteca, clique duas vezes em um operador personalizado.
2. Na barra de ferramentas, clique em Reservar.
3. Na janela do criador de operadores personalizados, clique na guia Formulário.
4. No painel Elementos do formulário, expanda todos os seguintes:
  - a. Elementos do formulário
  - b. Layout da página e todas as páginas abaixo de seu operador personalizado
5. Arraste um elemento de formulário a partir de Elementos do formulário até a página de seu operador personalizado. Não é possível arrastar um campo diretamente para o layout do formulário.
6. Clique no novo campo para editar suas propriedades. Renomeie o novo campo com um identificador significativo usando a propriedade Rótulo. Use o botão Renomear da barra de ferramentas para renomear determinados campos, como caixas de seleção.

Cada nome de campo corresponde a um campo nomeado na paleta Propriedades do operador personalizado do Criador de processos.
7. A ordem dos campos na paleta Propriedades também corresponde à ordem exibida na guia Formulário. Selecione um elemento e clique em Mover para cima ou Mover para baixo na barra de ferramentas para movê-lo para cima ou para baixo na lista.

### Para remover um parâmetro:

1. Selecione a página.
2. Na barra de ferramentas, clique em Remover.

## Opção Parâmetro invisível

O CA Process Automation permite configurar e ocultar os parâmetros do operador personalizado do usuário final. É possível definir os parâmetros do operador personalizado como invisíveis, somente leitura ou editáveis em tempo de execução. Para passar informações ao parâmetro do operador personalizado, adicione macros ao operador personalizado. É possível também ocultar parâmetros do usuário e usar macros para acessar os valores de entrada dos parâmetros ocultos do operador personalizado. As alterações feitas nos parâmetros do operador personalizado produzem um efeito em cascata, atingindo para baixo todos os processos que utilizam o operador personalizado.

A menos que você defina suas propriedades Somente leitura, Desativado ou Oculto como verdadeiro, os parâmetros personalizados adicionados ao operador personalizado serão visíveis e configuráveis pelos usuários finais na paleta Propriedades do operador personalizado no Criador de processos ou no Editor de programações.

## Propriedade Expandir macro no valor

Para usar um parâmetro personalizado e definir o valor de um parâmetro do operador base (que é o uso típico dos parâmetros personalizados), você deve tratá-lo como uma macro. Use a propriedade *Expandir macro no valor* para permitir que a entrada do usuário para um campo do operador personalizado seja usada como o valor do parâmetro do operador de base.

Para fazer isso, defina a propriedade Oculto primeiro. Em seguida, configure o parâmetro do operador base com a opção Expandir macro no valor. O CA Process Automation pesquisa os valores do parâmetro do operador base (para qualquer parâmetro com a opção Expandir macro no valor definida) para qualquer nome de parâmetro personalizado e substitui o nome do parâmetro personalizado pelo valor de parâmetro do cliente. Embora esse seja um recurso eficiente, tome cuidado ao determinar quais parâmetros do operador base devem ter a opção Expandir macro no valor definida e ao nomear seus parâmetros personalizados, para que eles sejam exclusivos o suficiente a fim de impedir que você substitua acidentalmente uma sequência de caracteres pelo parâmetro personalizado que compartilha o mesmo nome.

Por exemplo, você criou um operador personalizado usando o operador *Executar o script* como base. O script é definido para passar alguns parâmetros com base na entrada fornecida por um usuário do operador personalizado. Um campo de formulário é adicionado para obter essa entrada do usuário final, e ele é chamado *for*. No operador de base, adicione um parâmetro, insira o valor *for* e defina a propriedade *Expandir macro no valor* como *true*. Isso transmitirá corretamente a entrada do usuário configurada no campo de formulário do operador personalizado para o script como um parâmetro.

Se um usuário inserir o valor *Steve* para esse parâmetro personalizado, o script receberá um parâmetro com o valor *Steve*. No entanto, se você definir a propriedade *Expandir macro no valor* para o parâmetro do operador de base *inLineScript*, isso substituirá a palavra *for* em qualquer local do script pela palavra *Steve*. Isso é indesejável, pois o código de script para qualquer loop *Para* seria acidentalmente substituído por um termo inesperado, resultando em um erro de sintaxe no seu script.

### Mais informações:

[Operador personalizado: guia Propriedades](#) (na página 185)

## Testar a interface do operador personalizado

Depois de adicionar e configurar as páginas e os parâmetros, é possível usar o recurso *Testar* para visualizar o painel Propriedades do operador personalizado.

**Siga estas etapas:**

1. Clique em Testar na barra de ferramentas.

Os parâmetros são exibidos nas guias na mesma ordem listada nas páginas correspondentes da paleta Parâmetros do operador personalizado.

2. Revise as páginas e os parâmetros.

Você pode visualizar a exibição de usuário final do operador do cliente e as opções disponíveis para ela.

**Exemplo: Recuperar valores da matriz Valuemap com uma função do sistema de operador**

Esse exemplo usa a função de sistema de operador a seguir para recuperar os valores de uma matriz de mapa de valores. A função retorna um campo ou uma coluna de uma matriz que tem como base os parâmetros fornecidos.

```
getValueFromArray(groupName, arrName, fieldName,
fieldName, requiredFieldName)
```

**Siga estas etapas:**

1. Crie um operador personalizado chamado CustOp\_GetDBVersion com Get Version como o operador de base.
2. Crie o formulário conforme mostra a ilustração a seguir:

The screenshot displays a software interface for configuring a custom operator. The interface is divided into two main panes. The left pane, titled 'Elementos do formulário', shows a tree view of the form structure. The right pane shows the visual representation of the form fields.

**Form Structure (Left Pane):**

- Elementos do formulário
  - CustOp\_GetDBVersion
    - Page Layout
      - Page
        - NamedConnection
          - Named Connection
          - HostName
          - UserName
          - Password
          - Port
          - Database Name
          - Database Type
          - Driver Name

**Form Fields (Right Pane):**

The form fields are organized into a section titled 'NamedConnection'. The fields are:

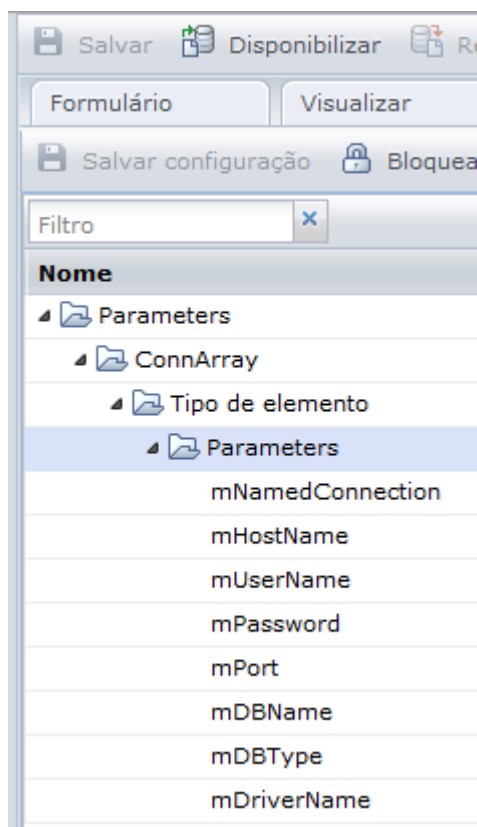
- Named Connection
- Host Name
- UserName
- Password
- Port
- Database Name
- Database Type
- Driver Name

Nomeie os elementos do formulário como `mNamedConnection`, `mHostName`, `mUserName`, `mPassword`, `mPort`, `mDBName`, `mDBType` e `mDriverName`. Certifique-se de que os nomes de elementos do formulário são semelhantes aos nomes de variáveis definidos na guia Configuração do grupo.

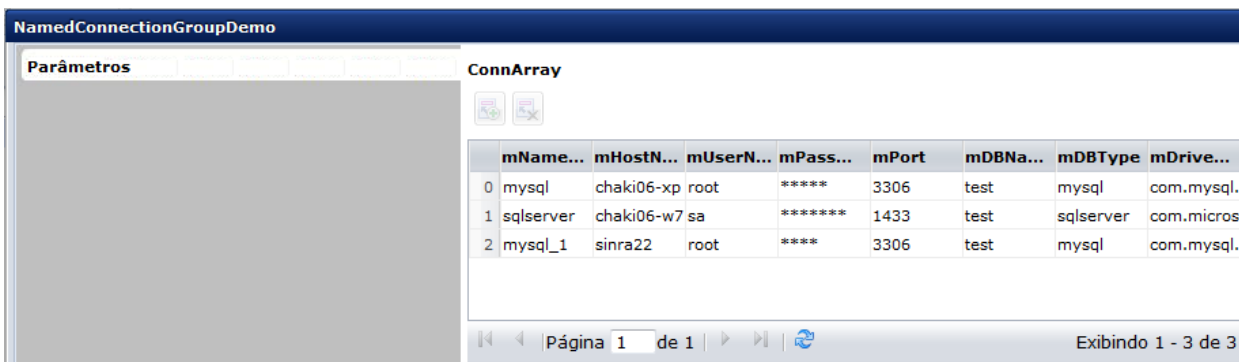
3. Clique na guia Configurações e crie um grupo chamado `NamedConnectionGroupDemo`.
4. Clique na guia Execução anterior do operador personalizado e insira o seguinte código:

```
Process.HostName =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mHostName");  
Process.UserName =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mUserName");  
Process.Password =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mPassword");  
Process.Port =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mPort");  
Process.DBName =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mDBName");  
Process.DBType =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mDBType");  
Process.DriverName =  
getValueFromArray("NamedConnectionGroupDemo",  
"ConnArray", "mNamedConnection", Operator.mNamedConnection,  
"mDriverName");
```

5. Selecione a guia Configuração do grupo e clique em Bloquear para bloquear o grupo.
6. Crie uma matriz de mapa de valores chamada `ConnArray` e adicione os parâmetros mostrados na ilustração a seguir:



7. Clique em Salvar configuração para salvar a configuração do grupo.
8. Clique em Desbloquear para publicar o grupo NamedConnectionGroupDemo no nível do domínio e do ambiente.
9. Abra o grupo na guia Módulos do Navegador de configuração.
10. Adicione os valores de parâmetro à matriz, conforme mostrado na ilustração a seguir, e salve a matriz.



Após os valores da matriz de mapa de valores serem recuperados, um usuário de produção poderá usar o operador NamedConnectionGroupDemo em um processo para fazer referência a uma determinada conexão.

1. Crie um processo que usa o operador personalizado CustOp\_GetDBVersion.
2. Forneça "mysql" como o valor do campo de conexão nas propriedades do operador personalizado para recuperar os valores relacionados da matriz ConArray.
3. Executar o processo.

O script do código de execução anterior do operador personalizado é executado. O processo recupera os valores para o valor do campo de conexão "mysql" inserido na propriedade de conexão nomeada e os exibe na paleta Conjunto de dados.

### **Operador personalizado: guia Visualizar**

Use a guia Visualizar para testar os elementos do formulário.

## Operador personalizado: guia Configurações

Use a guia Configurações para definir as configurações que são comuns a todos os operadores. A guia Configurações do operador personalizado inclui as propriedades do operador de base padrão. Estas configurações correspondem aos parâmetros encontrados na paleta Propriedades do Criador de processos.

### **Destino**

Define o local em que o operador personalizado é executado.

### **O destino é uma expressão calculada**

Indica que o destino usa uma expressão calculada.

### **Destino somente leitura**

Indica que o destino é somente leitura e não pode ser alterado.

### **Executar como usuário chamador**

Indica que o operador é executado como se fosse a entidade de chamada.

### **'Executar como usuário chamador' somente leitura**

Indica que o operador é executado como se fosse a entidade de chamada, mas como somente leitura, de forma que ele não pode ser alterado.

### **Grupo**

Especifica o nome de um grupo para o operador personalizado. Essa configuração é usada como o título do grupo ou da pasta na paleta Operadores. É possível usar o mesmo nome de grupo para os operadores personalizados relacionados, de modo que todos eles apareçam na mesma pasta na paleta Operadores. A configuração do Grupo de operadores personalizados define parâmetros e valores comuns para operadores personalizados no grupo.

### **Nome de exibição**

Indica o nome que é exibido na paleta Operadores com o ícone do seu operador personalizado. O nome deve ser curto e se basear na função do seu operador. O nome de exibição também é usado para fornecer o valor inicial para o campo Nome na página Informações das Propriedades do operador. É possível usar qualquer combinação de letras, números, espaços e caracteres de sublinhado.

#### **Execução anterior do operador personalizado**

Especifica qualquer código que deve ser executado antes que o operador personalizado seja executado.

#### **Execução posterior do operador personalizado**

Especifica qualquer código que deve ser executado depois que o operador personalizado for executado.

#### **Ícone da exibição atual**

Especifica o ícone que representa o operador. Por padrão, o ícone do seu operador personalizado é o ícone do operador base.

### **Criar um grupo de operadores personalizados**

Os criadores de conteúdo ou administradores podem usar a guia Configurações para criar um grupo de operadores personalizados. Somente os administradores podem, em seguida, configurar o grupo de operadores personalizados na guia Configuração de grupo.

#### **Siga estas etapas:**

1. Clique na guia Biblioteca e vá até a pasta que contém o operador personalizado no qual deseja basear o grupo de operadores personalizados.
2. Clique duas vezes no operador personalizado de destino.
3. Clique na guia Configurações.
4. Digite o nome do grupo de operadores personalizados no campo Grupo e, em seguida, clique em Salvar.

### **Atribuir o grupo a outros operadores personalizados**

Os criadores de conteúdo que estiverem adicionando operadores personalizados poderão usar a guia Configurações para atribuir um grupo adequado ao operador personalizado.

#### **Siga estas etapas:**

1. Para abrir o editor de operador personalizado, clique duas vezes em um operador personalizado na guia Biblioteca.
2. No editor de operador personalizado, clique na guia Configurações.
3. Na lista suspensa Grupo, selecione o nome do grupo.
4. Clique em Salvar.

## Código de execução anterior e de execução posterior específico ao operador personalizado

O CA Process Automation permite definir o código JavaScript de execução anterior e posterior de operadores personalizados. O código de execução anterior é processado antes de um operador ser executado, e o código de execução posterior é processado após um operador ser executado. Os usuários de operador personalizado não podem substituir o código de execução anterior ou posterior que os criadores ou desenvolvedores que possuem mais permissões já definiram.

O código é executado na seguinte ordem:

1. Código de execução anterior definido pelo usuário.
2. Código de execução anterior específico ao operador personalizado.
3. O operador personalizado.
4. Código de execução posterior específico ao operador personalizado.
5. Código de execução posterior definido pelo usuário.

*Definir código de execução anterior e posterior específico para o operador personalizado*

Para impedir a exclusão ou modificação acidental de um código de execução, é possível definir um código específico para cada operador personalizado. Você também pode usar os dados de entrada ou de saída do operador personalizado no seu código.

### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique duas vezes em um operador personalizado.  
A caixa de diálogo Operador personalizado é exibida.
3. Clique na guia Configurações.
4. Clique em um dos dois botões a seguir:
  - Clique em Execução anterior do operador personalizado para inserir o código de execução anterior.
  - Clique em Execução posterior do operador personalizado para inserir o código de execução posterior.

Uma caixa de diálogo de código aparece.

5. Insira seu código. Clique em OK.
6. Clique em Salvar.

*Ordem de execução para código de execução anterior específico ao operador personalizado*

A ordem de execução para o código de execução anterior específico ao operador personalizado é a seguinte:

1. Executar o código de execução anterior definido pelo usuário (ou seja, o código de execução anterior definido no processo).
2. Avaliar os Parâmetros de usuário (esta etapa inclui a avaliação de expressões e a expansão de macros).
3. Executar o código de execução anterior definido personalizado. Os parâmetros de usuário são expostos e você pode usar a palavra-chave Operador para acessar os parâmetros.

**Observação:** o designer não tem as permissões para alterar valores do parâmetro de usuário.

4. Avaliar os Parâmetros padrão (esta etapa inclui a avaliação de expressões e a expansão de macros).
5. Avaliar o operador de base.

### Operador personalizado: guia Conjunto de dados

Para definir e agrupar variáveis do conjunto de dados do operador que contenham informações retornadas por um operador personalizado, use a guia Conjunto de dados.

Por exemplo, suponha que o seu operador personalizado recupere campos de um ticket em um sistema de emissão de tickets interno. É possível criar uma página para agrupar todos os campos recuperados e colocar os parâmetros na página correspondente aos campos retornados.

Essa maneira de usar a guia Conjunto de dados fornece os seguintes benefícios:

- Os usuários do operador personalizado podem ver facilmente como a saída é definida sem precisar executar o operador primeiro para criar os parâmetros em tempo de execução.
- É possível agrupar parâmetros relacionados em uma página com um nome descritivo, o que não é possível em tempo de execução.

Na paleta Conjunto de dados é possível adicionar, renomear, excluir e mover páginas e parâmetros. Também é possível configurar as definições do parâmetro para definir o tipo, o valor inicial e outras características. As configurações e as definições na paleta Conjunto de dados de um operador personalizado são as mesmas para qualquer outro conjunto de dados.

Para cada par de nome/valor do parâmetro de saída à esquerda, você pode configurar se deseja ocultar o valor do parâmetro de saída.

#### Ocultar da saída

Especifica se o parâmetro deve ser incluído na saída.

- Marcada: indica que o parâmetro deve ser oculto; o parâmetro de saída não é exibido no conjunto de dados.
- Desmarcada: indica que o parâmetro deve ser incluído nos parâmetros de saída; o conjunto de dados inclui o parâmetro de saída.

#### Mais informações:

[Conjuntos de dados](#) (na página 203)

## Operador personalizado: guia Configuração do grupo

Na guia Configurações, os criadores de conteúdo ou administradores de conteúdo podem:

- [Criar um grupo de operadores personalizados](#) (na página 178) para um operador personalizado.
- [Atribuir o grupo a outros operadores personalizados](#) (na página 178).

Na guia Configuração do grupo, os administradores de conteúdo podem:

- [Configurar um grupo de operadores personalizados](#) (na página 182).
- [Editar um grupo de operadores personalizados](#) (na página 183).

Na guia Módulos do Navegador de configuração, os administradores adicionam valores padrão para as variáveis configuradas, que então são herdadas. A herança procede do Domínio para os ambientes, e de cada ambiente para os operadores que são executados no ambiente. Os administradores podem substituir os valores do nível Domínio no nível de ambiente. Os criadores de conteúdo podem substituir as configurações do nível de ambiente no nível de operador personalizado.

## Configurar e publicar um grupo de operadores personalizados

Um grupo de operadores personalizados define os parâmetros que são comuns ao grupo de operadores personalizados. Os criadores de conteúdo (ou administradores) usam a guia Configurações para criar um grupo de operadores personalizados. Em seguida, os administradores poderão usar a guia Configuração do grupo para adicionar as variáveis apropriadas e publicar o grupo para o domínio e seus ambientes. A publicação adiciona o grupo de operadores personalizados na guia Módulos no Navegador de configuração.

**Observação:** depois de publicar as variáveis, configure seus valores na guia Configuração. Os operadores personalizados herdam automaticamente os valores padrão do grupo de operadores personalizados ao qual o operador personalizado pertence.

### Siga estas etapas:

1. Para abrir o editor de operador personalizado, clique duas vezes em um operador personalizado na guia Biblioteca.
2. No editor de operador personalizado, clique na guia Configuração do grupo.

**Observação:** não é necessário reservar o operador personalizado para configurar um grupo.

3. Clique em Bloquear para bloquear o grupo.

As opções Adicionar variável e Adicionar página são ativadas.

4. Adicione variáveis ou páginas, conforme apropriado.
5. Clique em Salvar configuração.
6. Clique em Desbloquear.
7. Clique em OK na mensagem a seguir:

As variáveis da configuração do grupo foram publicadas para o domínio e o ambiente.

O CA Process Automation inclui o novo grupo de operadores personalizados na guia Módulos do domínio e de cada ambiente no domínio.

Na guia Módulos, no Navegador de configuração, os administradores podem abrir o grupo de operadores personalizados que você definiu na guia Configurações. Em um grupo de operadores personalizados aberto, os administradores podem adicionar valores para as variáveis definidas por você na guia Configuração do grupo.

**Observação:** para obter mais informações sobre como configurar valores para grupos de operadores personalizados, consulte o *Guia do Administrador de Conteúdo*.

## Editar a configuração do grupo de operadores personalizados

Os administradores podem editar a configuração de um grupo de operadores personalizados. A republicação não atualiza todas as alterações que você faz na guia Módulos do Navegador de configuração. Se um grupo que você editar já tiver sido publicado pelo menos uma vez ou se outro usuário estiver usando o grupo, o CA Process Automation publicará apenas os parâmetros novos para o grupo. As operações de exclusão ou alteração do tipo de dados em variáveis que já foram publicadas não são atualizadas no nível de Domínio ou de ambiente quando você republica. No entanto, o CA Process Automation salva as alterações no nível do operador personalizado. É possível exibir as variáveis de tipo de dados alteradas ao abrir o operador personalizado em um grupo.

### Siga estas etapas:

1. Para abrir o editor de operador personalizado, clique duas vezes em um operador personalizado na guia Biblioteca.
2. Clique na guia Configuração do grupo.
3. Clique em Bloquear para bloquear o grupo.

O CA Process Automation permite as opções Adicionar variável e Adicionar página.

4. Adicione, exclua ou edite as variáveis ou as páginas, conforme necessário.
5. Clique em Salvar configuração.
6. Clique em Desbloquear.

Quando você usar um operador personalizado no grupo editado na guia Criador, verifique se o operador personalizado inclui todas as suas alterações do grupo (incluindo alterações no tipo de dados).

## Operador personalizado: guia Painéis personalizados

A guia Painéis personalizados é exibida apenas para um subconjunto limitado de operadores de base, como os operadores *Atribuir tarefa de usuário* ou *Solicitar método SOAP*. As seções na guia variam de acordo com o operador de base. Dois operadores de base comuns são exibidos nos exemplos a seguir.

Para operadores personalizados com base no operador *Atribuir tarefa de usuário*, a guia Painéis personalizados está dividida nas seguintes seções:

### Responsáveis

Especifica os usuários ou grupos que podem interagir com o operador personalizado e seu formulário.

### Transferir/delegar filtros

Especifica os usuários ou grupos que estão disponíveis para a transferência ou a delegação da tarefa.

### Tarefa de usuário

Especifica o título, a descrição e o código de inicialização de dados do formulário para o Formulário de solicitação de interação associado.

Para operadores personalizados com base no operador *Solicitar método SOAP*, a guia Painéis personalizados está dividida nas duas seções a seguir:

### Dados de chamada SOAP

Use esse grupo de campos para especificar o URL do serviço, o nome do método, o nome do usuário, a senha, a versão, a origem e outros detalhes da autenticação HTTP ou SOAP básica. Use o Assistente do WSDL para carregar um URL e selecionar serviços, portas e operações do WSDL.

### Parâmetros dinâmicos

Use esse grupo de campos para especificar o estilo do parâmetro e adicionar, editar ou excluir macros ou consultas XPath.

### Resultados da chamada

Use esse grupo de campos para especificar o caminho do arquivo de resposta da solicitação de SOAP e adicionar, editar ou excluir mais dados extraídos. Você também pode marcar as opções para determinar como partes do corpo, cabeçalho e espaços para nome XML de SOAP extraídas serão manipuladas.

### Anexo MIME

Use esse grupo de campos para especificar uma expressão ou adicionar, editar e excluir conteúdo como anexos MIME.

### Segurança do WS

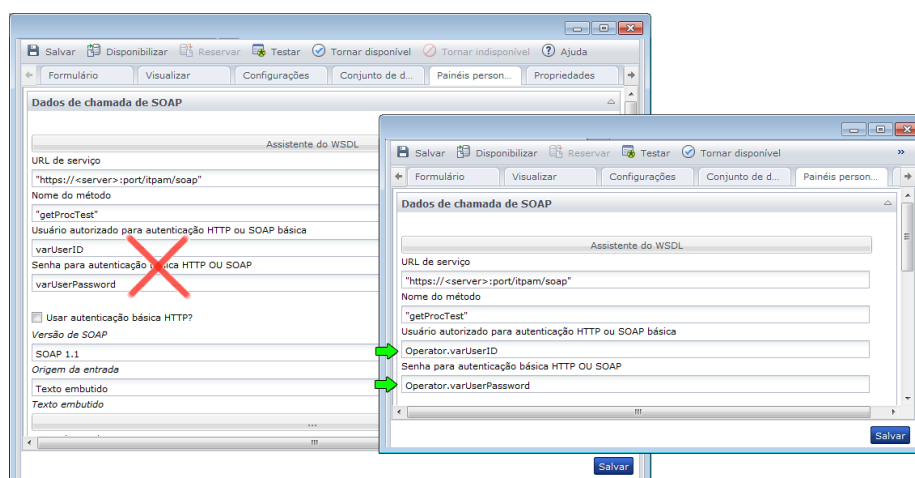
Use esse grande grupo de campos para gerenciar detalhes da segurança. Amplie cada caixa do grupo clicando no respectivo título para exibir todos os campos.

## Sintaxe de expansão de macro

Em versões anteriores do CA Process Automation, era possível usar uma variável em um campo de operador personalizado que aceitasse uma expressão como entrada. O operador de base oferecia suporte ao uso de variáveis e as substituía dinamicamente por valores reais, um conceito que é conhecido como *expansão de macro*. Os operadores personalizados que você importar de versões anteriores continuarão a dar suporte a variáveis dinâmicas e à expansão de macro nesta release do CA Process Automation.

Entretanto, para determinados operadores de base, especialmente aqueles com uma guia Painéis personalizados, os campos não oferecem suporte à expansão de macro com apenas um nome de variável. Exemplos incluem os operadores Atribuir tarefa de usuário e Solicitar método SOAP. Adicione o termo *operador* no campo para avaliação da expressão e da expansão de macro, se houver suporte. A imagem a seguir demonstra a sintaxe necessária para substituir as variáveis *varUserID* e *varUserPassword* dinamicamente:

```
Operator.varUserID  
Operator.varUserPassword
```



## Operador personalizado: guia Propriedades

Use essa guia para armazenar as tags de nome, descrição e palavras-chave de seu operador personalizado. As propriedades básicas nessa guia funcionam da mesma maneira para todos os objetos de automação.

### Operador personalizado: guia Versões

Use essa guia para gerenciar as versões do operador personalizado. As versões funcionam da mesma forma para todos os objetos de automação.

### Operador personalizado: guia Trilha de auditoria

Use essa guia para examinar o histórico do seu operador personalizado. As Trilha de auditoria funcionam da mesma forma para todos os objetos de automação.

### Definir a disponibilidade do operador personalizado para todos os usuários

Por padrão, um operador personalizado não está disponível para outros criadores do CA Process Automation. É possível definir a disponibilidade de operadores personalizados para todos os usuários.

A disponibilidade de um operador personalizado afeta apenas a capacidade de ver e adicionar o operador personalizado a um processo ou programação. Depois que um operador personalizado tiver sido adicionado a um processo ou uma programação, torná-lo indisponível não afeta seu uso e inclusão existentes.

#### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique duas vezes em um operador personalizado.  
A caixa de diálogo Operador personalizado é exibida.
3. Na barra de ferramentas:
  - a. Clique em Tornar disponível para permitir que todos os usuários vejam e usem o operador.
  - b. Clique em Tornar indisponível para ocultar o operador de outros usuários.

A alteração na disponibilidade é instantânea, clicando ou não em Salvar.

## Publicar a configuração de um grupo de operadores personalizados em outro domínio

Quando você define um grupo de operadores personalizados, ele é automaticamente publicado no domínio atual e em todos os ambientes do domínio.

Quando você importa para outro domínio, o processo permite publicar o grupo de operadores personalizados nesse domínio e em todos os seus ambientes. A opção Publicar a configuração do grupo do operador personalizado requer direitos de Group\_Config\_Admin. Os administradores (integrantes do grupo PAMAdmins) possuem esse direito.

### Siga estas etapas:

1. No navegador da biblioteca para o orquestrador selecionado, selecione o destino para os itens importados.
2. Clique com o botão direito do mouse na pasta pai e, em seguida, clique em Importar.
3. Clique em Procurar, procure o arquivo XML na janela Abrir e clique em Abrir.
4. Selecione uma das opções a seguir para especificar como tratar os objetos com conflitos de nomes. Se você importar para uma pasta vazia, não haverá conflito.
  - Importar
  - Importar e substituir
  - Não importar
5. (Opcional) Marque a caixa de seleção Definir a versão importada como atual. Essa opção será aplicada se você selecionar *Importar como uma nova versão e manter o objeto existente* na Etapa 4.
6. Marque a caixa de seleção Tornar os operadores personalizados disponíveis. Não é possível usar operadores personalizados até que eles estejam disponíveis.
7. Marque a caixa de seleção Publicar a configuração do grupo do operador personalizado.

A publicação publica os grupos de operadores personalizados no domínio de importação e em todos os ambientes desse domínio.
8. Clique em Enviar.
9. Clique em OK na mensagem de confirmação de importação bem-sucedida.

A importação recria a estrutura de pastas exportada no local selecionado, importa os operadores personalizados e publica o grupo de operadores personalizados e as variáveis relacionadas.

## Usando os operadores personalizados

É possível usar um operador personalizado em um processo ou uma programação como qualquer outro operador. Lembre-se dos seguintes pontos ao trabalhar com operadores personalizados:

- Os valores iniciais para os parâmetros do operador personalizado podem ser pré-configurados ou ocultos.
- Talvez existam novas entradas do parâmetro personalizado que exijam configuração.
- O operador personalizado será exibido em uma pasta chamada Não categorizado, a menos que você tenha definido um nome de grupo na guia Configurações da caixa de diálogo Operador personalizado.
- Se um operador personalizado não estava disponível no momento em que seu processo foi aberto para edição, torne-o disponível e atualize a pasta do grupo de operadores novamente.
- Se um operador personalizado não estava disponível no momento em que sua programação foi aberta para edição, torne-o disponível e atualize a pasta do grupo de operadores novamente.
- Você deve atualizar a pasta do grupo de operadores em um processo ou uma programação para ver as alterações feitas no nome ou em outras configurações do operador personalizado.
- É preciso fechar e reabrir um processo ou uma programação para ver as alterações feitas nos operadores personalizados existentes.
- Qualquer processo ou programação usando um operador personalizado usa a última versão disponibilizada marcada como *Atual* no Navegador da biblioteca.

## Editar valores do operador personalizado

Os grupos de operadores personalizados fornecem variáveis personalizadas com valores padrão. Os operadores personalizados que você atribuir a um grupo de operadores personalizados compartilham essas variáveis e esses valores. Você pode aceitar os valores no nível do grupo ou pode atribuir valores diferentes para as variáveis para um operador personalizado.

**Observação:** os valores do grupo são definidos no nível de Domínio. Todos os ambientes no Domínio herdam esses valores do grupo. Ao contrário das categorias de operador, não é possível substituir valores de grupos de operadores personalizados no nível do orquestrador ou do agente. No entanto, é possível alterar os valores herdados no nível do operador personalizado.

### Siga estas etapas:

1. Clique na guia Criador.
2. Clique na lista suspensa Exibir e selecione Operadores, Conjunto de dados e Propriedades.
3. Expanda o nome do grupo de operadores personalizados na lista Operadores e arraste o operador personalizado selecionado para a tela.
4. Clique duas vezes no operador personalizado.
5. Verifique o valor de cada variável personalizada. Aceite o valor herdado ou digite o valor desejado.
6. Clique em Salvar.

## Como trabalhar com operadores personalizados protegidos

O CA Technologies pode liberar operadores personalizados protegidos. O ícone de cadeado indica que o operador personalizado está protegido:



O uso de operadores personalizados protegidos é restrito nos seguintes aspectos:

- Não é possível exibir nenhum código definido para o operador personalizado protegido. Os parâmetros de entrada, o código de execução anterior e o código de execução posterior são criptografados.
- Não é possível modificar a versão importada.
- Não é possível modificar nem exibir o código em uma cópia da versão importada.

## Seus operadores favoritos

Por motivos de conveniência, você pode adicionar seus operadores favoritos à pasta ou ao grupo Favoritos. O grupo Favoritos é exibido na parte superior da paleta Operadores no Criador de processos.

Clique em Atualizar na paleta Operadores para exibir seus operadores favoritos ao criar um processo.

## Adicionar ou remover seus operadores favoritos

Você pode adicionar e remover seus operadores favoritos na pasta do grupo Favoritos da paleta Operadores.

### Siga estas etapas:

1. Clique na guia Criador.
2. Abra um processo ou crie um.
3. Na barra de ferramentas, clique no menu Exibir e selecione Operadores.  
A paleta Operadores é exibida.
4. Na área de filtro na parte superior da paleta Operadores, clique em Atualizar.  
A lista de operadores, operadores personalizados e operadores favoritos é atualizada.
5. Expanda a pasta Favoritos para exibir os operadores adicionados. Para novos usuários, os operadores são exibidos.
6. Para adicionar um operador à pasta Favoritos:
  - a. Expanda qualquer outra pasta de operadores.
  - b. Clique com o botão direito do mouse em um operador e selecione Adicionar aos favoritos.
  - c. Clique em Atualizar.
7. Para remover um operador da pasta Favoritos:
  - a. Clique em Atualizar.
  - b. Expanda a pasta Favoritos para exibir os operadores adicionados.
  - c. Clique com o botão direito do mouse em um operador e selecione Remover de favoritos.

## Conectores

*Conectores* são extensões opcionais para o CA Process Automation que permitem a interface dos operadores com outras soluções da CA e de terceiros. Quando o administrador configura um conector, um novo grupo de operadores é exibido na paleta Operadores no Criador de processos.

Os conectores integram produtos da CA e de terceiros nos processos de fluxo de trabalho e fornecem serviços de ponte de outros produtos para os produtos e as soluções da CA que incorporam o CA Catalyst. Você pode fazer download de outros conectores além daqueles fornecidos na mídia de instalação do CA Process Automation. Restrições de licenciamento podem ser aplicadas. Veja uma lista dos conectores mais conhecidos:

- Amazon Web Services (AWS)
- BMC Remedy
- CA Client Automation
- CA CMDB
- CA Configuration Automation
- CA eHealth
- CA NSM
- CA Service Desk Manager
- CA Spectrum IM
- CA Workload Automation AE
- IBM AS/400
- IBM z/OS
- Microsoft Hyper-V
- VMware vSphere

Use conectores específicos ao aplicativo para executar tarefas, como coleta de dados ou aplicação de ações, em sistemas e aplicativos de destino. Os conectores fornecem operadores executados nos seguintes locais:

- No orquestrador
- Em agentes que residem no servidor de aplicativos
- Em agentes do proxy que possam executar remotamente a tarefa necessária ou coletar os dados no servidor de aplicativos

Cada módulo do conector normalmente inclui vários operadores. Cada operador executa uma das seguintes tarefas especializadas:

### **Suporte à árvore de decisões**

Retorna um binário (verdadeiro/falso, êxito/falha) que pode ser usado para determinar a ramificação do processo. Em algumas situações, o valor retornado pode ter mais do que duas opções. No entanto, o valor retornado é sempre um conjunto pequeno e bem definido de valores possíveis.

### Coleta de dados

Coleta conjuntos de dados mais complexos do aplicativo de destino. O resultado geralmente é armazenado em um conjunto de dados local em que outros operadores podem analisar melhor. Se for necessário, os dados também podem ser tornados globais para que outros processos possam utilizá-los.

### Gerenciamento ativo

Executa ações no sistema de destino, incluindo todos os operadores que alteram o comportamento do sistema de destino de alguma maneira. Por exemplo:

- Envio de um evento
- Reconfiguração do aplicativo
- Início ou interrupção de um serviço relacionado

Em alguns casos, um único operador pode executar mais de uma dessas funções. Por exemplo, um conector pode ser executado e, em seguida, retornar um conjunto de resultados com base na ação do conector.

O CA Process Automation inclui muitos conectores. É possível criar conectores e operadores relacionados, com facilidade, de uma destas maneiras:

- Chamando executáveis padrão, específicos ao aplicativo, que estejam no agente
- Usando linguagens de script padrão para funções mais complexas

Para minimizar a pegada do aplicativo e a complexidade da interface do usuário, apenas um conjunto de conectores genérico e de uso comum é instalado por padrão. A melhor prática é instalar outros conectores somente quando necessário.

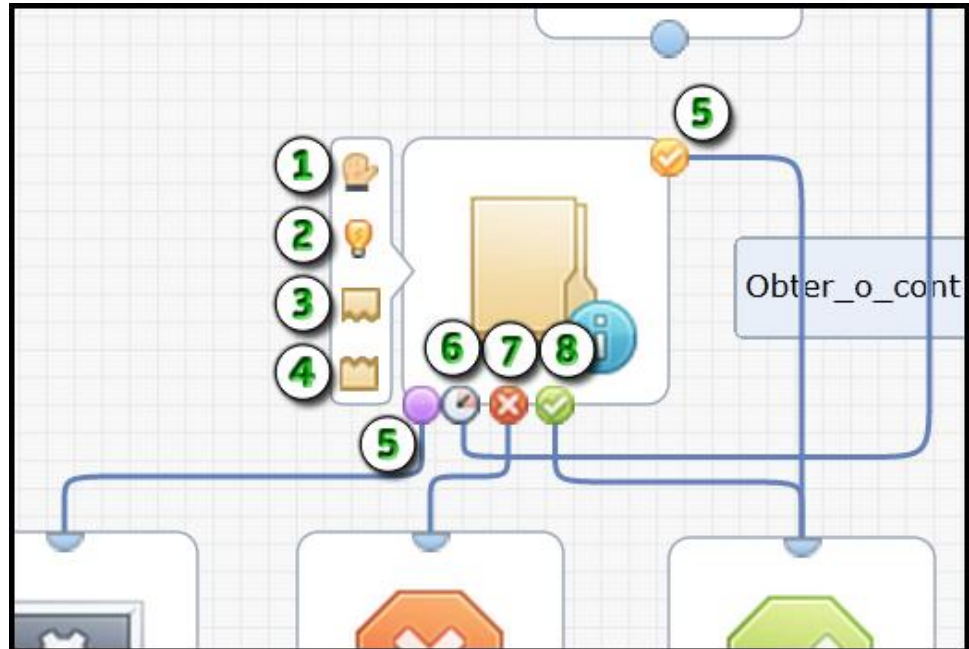
## Ícones dos operadores

O CA Process Automation fornece ícones padrão que serão exibidos para os operadores quando eles forem colocados em um processo ou uma programação. Você pode substituir o ícone padrão por um ícone que você especifica, (um *ícone personalizado*). É possível atribuir um ícone personalizado a qualquer operador em um processo, incluindo operadores personalizados.

O CA Process Automation manipula automaticamente os indicadores visuais menores que representam o status de um operador em um processo. Por exemplo, gráficos menores nos cantos de todos os ícones representam diferentes estados de execução (como ocioso, em execução, concluído e com falha) em um processo.

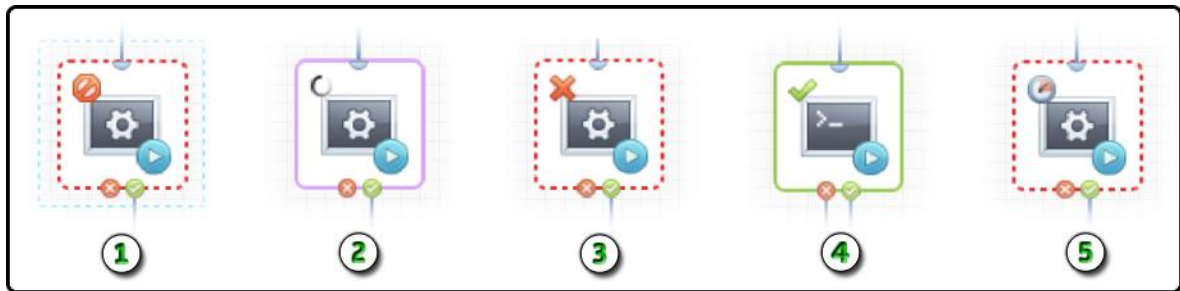
## Ícones de status do operador

O gráfico a seguir descreve o pequeno subconjunto de ícones que um operador pode exibir para indicar o status e as opções de porta.



Item:	Descrição:
1	<b>Ponto de interrupção:</b> este ícone indica que você definiu um ponto de interrupção no processo nesse operador. Clique em Definir ponto de interrupção na barra de ferramentas.
2	<b>Módulo de simulação:</b> este ícone indica a substituição das opções de simulação e a definição do tipo de simulação do operador para <i>local</i> ou <i>distante</i> . Na paleta Propriedades, expanda Simulação.
3	<b>Código de execução anterior:</b> este ícone indica a presença de código JavaScript que é executado antes do operador ser executado. Na paleta Propriedades, expanda Configurações de execução.
4	<b>Código de execução posterior:</b> este ícone indica a presença de código JavaScript que é executado depois do operador ser executado. Na paleta Propriedades, expanda Configurações de execução.
5	<b>Personalizar porta:</b> o processo flui por uma porta personalizada quando sua expressão predefinida é verdadeira.
6	<b>Porta Tempo limite:</b> o processo flui por essa porta quando o operador expira.
7	<b>Porta Com falha:</b> o processo flui por essa porta quando o operador gera um resultado sem êxito ou falha.
8	<b>Porta Concluído:</b> o processo flui por essa porta quando o operador gera um resultado com êxito.

Quando você executa um processo, os ícones a seguir indicam o status de cada operador.



Item:	Descrição:
1	<b>Cancelado:</b> o processo foi interrompido.
2	<b>Em execução:</b> o processo está em execução no momento.
3	<b>Com falha:</b> o processo falhou nesse operador específico.
4	<b>Concluído:</b> o processo passou com êxito nesse operador específico.
5	<b>Tempo limite:</b> o processo expirou nesse operador específico.

## Criando, editando e aplicando ícones personalizados

Os *ícones personalizados* são identificadores visuais para um operador. Eles ajudam você a identificar a função específica de um operador. É possível criar ícones personalizados em qualquer pasta da biblioteca de automação e, em seguida, aplicá-los a qualquer operador. Cada ícone exibe sua opção de imagem base com uma imagem de modificador sobreposta no canto inferior direito. A base (objeto) e o modificador (ação) fornecem uma estrutura consistente para todos os ícones.

Os tópicos desta seção descrevem por que e como um Criador de conteúdo do CA Process Automation usa ícones personalizados para personalizar a aparência dos operadores. Esta seção também fornece exemplos.

Como um criador de processos, você depende de indicações *visuais* fornecidas por ícones para determinar a finalidade de cada operador. Por exemplo, o que você imagina ser a função do seguinte ícone de operador?



Mesmo sem rótulo, é possível deduzir pela imagem que o ícone representa um operador *Excluir email*. Um ícone apropriado ajuda você e outros criadores a compreender a funcionalidade de um operador.

O investimento de tempo inicial dedicado à atribuição de um ícone apropriado trará vários benefícios, incluindo o seguinte:

#### Padronização

A utilização de um conjunto padrão de imagens de base e de modificador ajudará os criadores a compreender o objeto que está executando a ação.

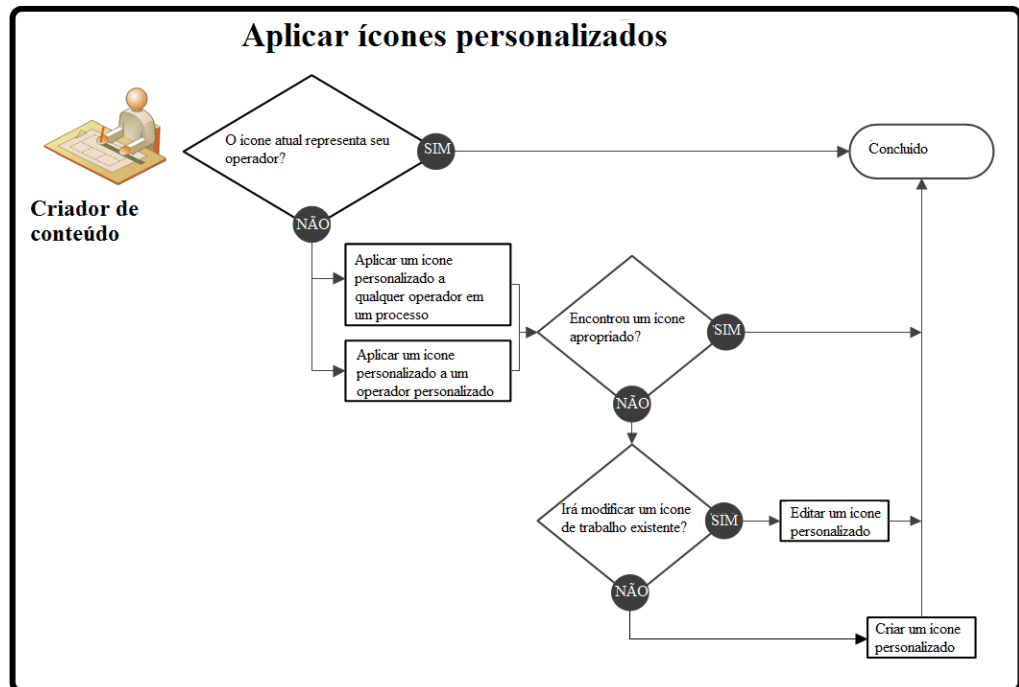
#### Simplicidade

Os criadores podem identificar prontamente operadores distintos até mesmo em processos maiores e mais complexos.

#### Compartilhamento

As equipes de criadores que estiverem trabalhando em outros idiomas nativos poderão compartilhar designs de processo, pois sua compreensão é mais fácil.

Use o fluxograma a seguir como guia ao personalizar ícones para seus operadores de processo:



Nenhuma sequência será necessária para executar as tarefas listadas. As opções a seguir sempre estarão disponíveis para você personalizar ícones:

- [Aplicar um ícone personalizado a qualquer operador em um processo](#) (na página 199): durante a criação de um processo, você pode alterar o ícone de qualquer operador para personalizar sua aparência para apenas uma única instância. Selecione uma instância específica de um operador e altere o respectivo ícone para um dos objetos predefinidos de ícone personalizado da biblioteca.
- [Aplicar um ícone personalizado a um operador personalizado](#) (na página 200): quando você altera um ícone de operador personalizado, sua escolha de ícone é aplicada a todas as futuras instâncias desse operador. Além disso, todos os criadores podem ver o novo ícone do operador personalizado na paleta Operador.
- [Editar um ícone personalizado](#) (na página 198): ao longo do tempo, você poderá editar um ou mais ícones personalizados predefinidos. Suas alterações serão aplicadas sempre que esse ícone personalizado já estiver sendo utilizado.
- [Criar um ícone personalizado](#) (na página 197): você pode definir um ou mais ícones personalizados. É possível criar uma série de ícones personalizados que você planeja atribuir a operadores ou operadores personalizados. Ou você pode simplesmente criar e salvar ícones personalizados sem saber com antecedência exatamente onde eles serão usados.

## Criar um ícone personalizado

É possível criar ícones personalizados em qualquer pasta da biblioteca de automação. Cada ícone emparelha sua opção de imagem base com uma imagem de modificador sobreposta no canto inferior direito da base. A base e o modificador fornecem uma estrutura consistente para todos os ícones.

### Siga estas etapas:

1. Clique na guia Biblioteca.  
Uma lista hierárquica de pastas é exibida no painel esquerdo, e todos os seus objetos de automação são exibidos na janela principal.
2. (Opcional) Se o novo ícone personalizado que você deseja criar for semelhante a um objeto de ícone personalizado existente, selecione-o, clique em Copiar e em Colar.  
Você agora pode editar a cópia para concluir seu ícone personalizado. Pule a próxima etapa.
3. Clique com o botão direito do mouse em qualquer pasta no painel Navegador da biblioteca e clique em Novo objeto, Ícone personalizado.  
O novo ícone é exibido no navegador com um nome padrão.
4. Clique duas vezes no ícone.  
O Editor de ícone personalizado é aberto
5. Selecione uma imagem base e uma imagem de modificador. Você pode procurar os ícones individualmente, filtrar pelo menu suspenso de categoria ou digitar uma palavra-chave de pesquisa.
6. Clique na guia Propriedades do objeto, digite ou edite o nome do ícone e, em seguida, clique em Salvar e fechar.  
O novo ícone personalizado está disponível na biblioteca.

Por exemplo, se quiser representar um operador denominado *Fazer upload do relatório*, você poderá:

- Selecionar uma imagem base que represente um relatório.
- Selecionar uma imagem de modificador que represente a ação de upload.

## Editar um ícone personalizado

Com o passar do tempo, os ícones personalizados criados por você e por outros criadores podem ser modificados para serem alinhados aos outros ícones. Você pode editar os ícones personalizados em qualquer pasta da biblioteca de automação. Cada ícone emparelha sua opção de imagem base com uma imagem de modificador sobreposta no canto inferior direito da base. A base e o modificador fornecem uma estrutura consistente para todos os ícones.

### Siga estas etapas:

1. Clique na guia Biblioteca.  
Uma lista hierárquica de pastas é exibida no painel esquerdo, e todos os seus objetos de automação são exibidos na janela principal.
2. Clique com o botão direito do mouse em um ícone personalizado.
3. Clique em Ação, Editar.  
O Editor de ícone personalizado é aberto.
4. Selecione uma imagem base e uma imagem de modificador, editar o nome do ícone e, em seguida, clique em Salvar.  
As modificações no ícone personalizado são aplicadas.

Por exemplo, se quiser representar um operador denominado *Fazer upload do relatório*, você poderá:

- Selecionar uma imagem base que represente um relatório.
- Selecionar uma imagem de modificador que represente a ação de upload.

## Aplicar um ícone personalizado a qualquer operador em um processo

Você pode alterar o ícone de qualquer operador utilizado em um processo.

### Siga estas etapas:

1. Clique na guia Criador.
2. Clique em Abrir.  
A caixa de diálogo Abrir processo é exibida.
3. Navegue até o seu processo e clique em Abrir.  
O processo que você abriu é exibido em uma nova guia.
4. Clique duas vezes em um operador específico já exibido na tela para mostrar suas propriedades. Na barra de ferramentas, clique no menu Exibir e selecione Propriedades para exibir a paleta Propriedades.
5. Na paleta Propriedades, expanda a seção Informações.
6. Na seção Informações:
  - a. Desmarque a caixa de seleção Usar ícone padrão.  
Uma borda é exibida em torno do ícone atual e o botão Procurar fica disponível.
  - b. Clique no botão Procurar.  
A caixa de diálogo Selecionar ícone personalizado é exibida.
  - c. Selecione o objeto de ícone personalizado que deseja usar para essa ocorrência específica do operador.
  - d. Clique em OK.
7. Na barra de ferramentas, clique em Salvar.

**Observação:** não é possível alterar os ícones associados ao estado de execução de um operador. Por exemplo, *em espera* ou *concluído*. O CA Process Automation gerencia automaticamente esses ícones.

## Aplicar um ícone personalizado a um operador personalizado

Você pode alterar o ícone para um operador personalizado. O ícone selecionado se aplica a todas as ocorrências futuras do operador personalizado em processos. As ocorrências existentes do operador personalizado em processos continuam a mostrar o ícone padrão original do operador de base.

### Siga estas etapas:

1. Clique na guia Biblioteca  
Uma lista hierárquica de pastas é exibida no painel esquerdo, e todos os seus objetos de automação são listados por tipo na janela principal.
2. Clique duas vezes em um operador personalizado.  
A caixa de diálogo Operador personalizado é exibida.
3. Na barra de ferramentas, clique em Reservar.
4. Clique na guia Configurações.
5. Na guia Configurações, clique no link Alterar o ícone.
6. Na caixa de diálogo Selecionar ícone personalizado, localize o ícone personalizado, selecione-o e clique em OK.
7. Clique em Salvar.
8. (Opcional) Se nenhum dos ícones existentes for adequado, edite um ícone personalizado ou crie outro.

**Observação:** não é possível alterar os ícones associados ao estado de execução de um operador personalizado. Por exemplo, *em espera* ou *concluído*. O CA Process Automation gerencia automaticamente esses ícones.

## Exemplos de ícones personalizados

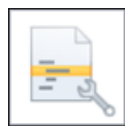
A lista a seguir mostra exemplos de ícones personalizados e os possíveis operadores que podem ser representados por eles.



Diagnosticar (ou monitorar) o desempenho



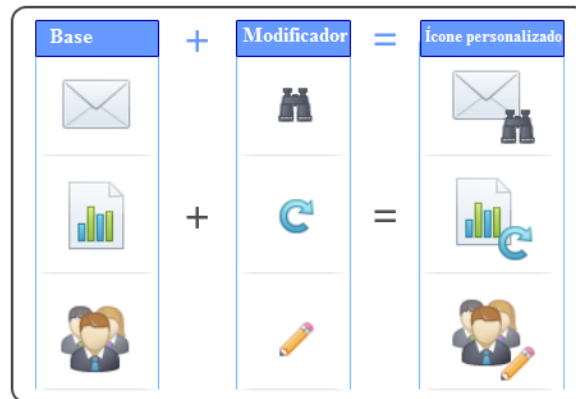
Adicionar conta de usuário



Depurar script ou código

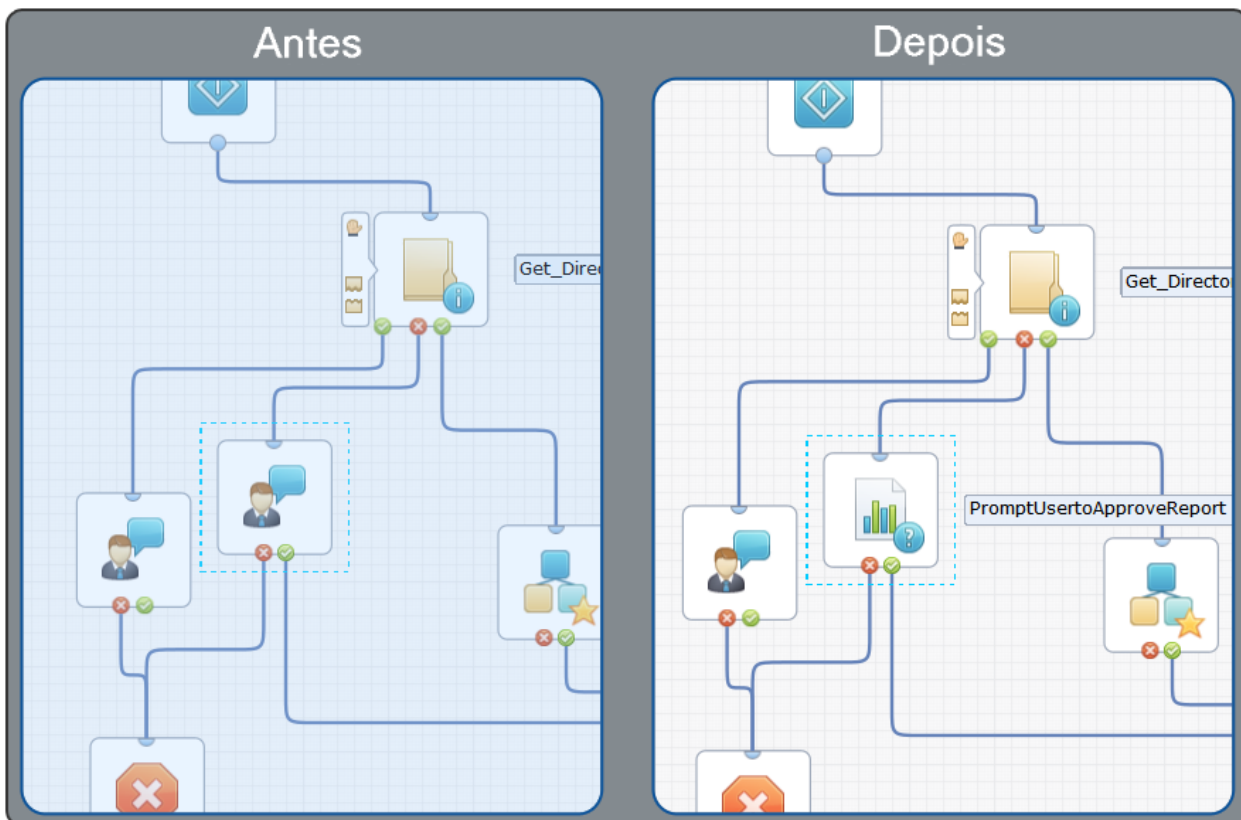
**Exemplo: Criar ou editar um ícone personalizado**

Este exemplo demonstra os principais conceitos do Editor de ícone personalizado do CA Process Automation. A imagem mostra a seleção de um ícone de base que é combinado a um ícone de modificador menor para criar um ícone personalizado. A combinação resultante é salvo como um objeto Ícone personalizado que pode ser aplicado posteriormente a qualquer operador.



### Exemplo: Aplicar um ícone personalizado

No exemplo a seguir, o criador decidiu alterar um de dois ícones idênticos. O operador personalizado baseia-se no operador Atribuir tarefa de usuário. No entanto, ele possui uma nova finalidade diferente. O operador solicita que um usuário aprove um relatório. O ícone padrão antigo é exibido no processo à esquerda (*Antes*) e o novo ícone personalizado é exibido à direita (*Depois*). O novo ícone representa melhor a função do operador atualmente selecionado no processo. Nesse exemplo, o criador também optou por mostrar o nome por extenso do operador.



Imagine um processo com vários operadores semelhantes organizados em uma série. Cada operador pode executar uma ação diferente. Nessa situação, use ícones personalizados para ajudá-lo a identificar cada operador distinto.

# Capítulo 6: Conjuntos de dados e parâmetros

---

Esta seção contém os seguintes tópicos:

[Conjuntos de dados](#) (na página 203)

[Parâmetros do processo](#) (na página 223)

## Conjuntos de dados

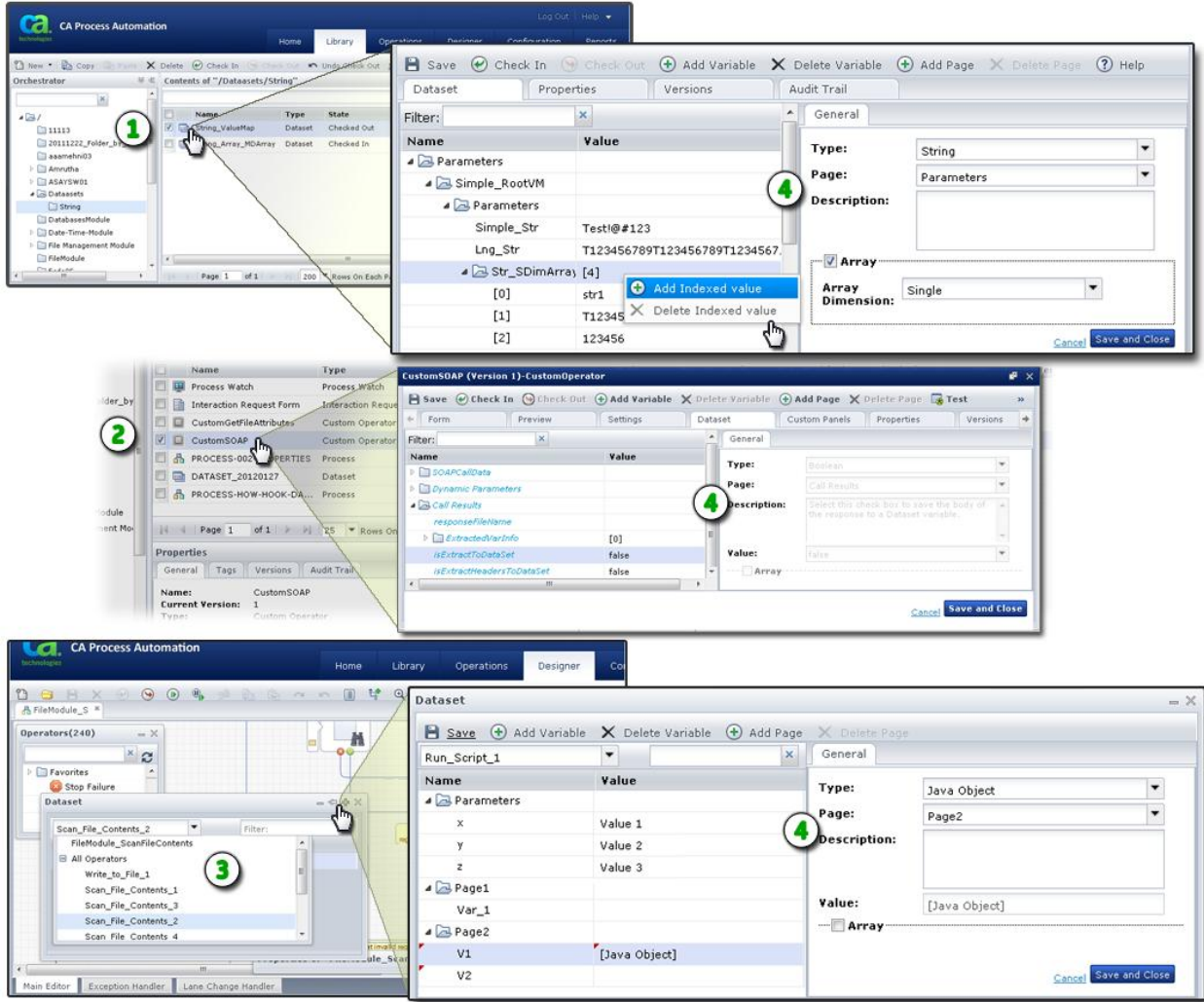
Os conjuntos de dados permitem definir grupos de variáveis para armazenar e organizar dados. Os conjuntos de dados fornecem uma maneira de compartilhar dados por várias instâncias do processo.

Um objeto *Conjunto de dados* define um conjunto de variáveis ao qual você pode fazer referência por nome. É possível criar e gerenciar conjuntos de dados no Navegador da biblioteca, assim como qualquer outro objeto de automação de um orquestrador. Um conjunto de dados pode conter qualquer número de campos, denominados *variáveis*. Atribua cada variável a um dos seguintes tipos de dados com base no tipo de dados armazenado por ela:

- *Booleano*
- *Data*
- *Sequência de caracteres*
- *Número inteiro*
- *Objeto Java*
- *Número longo*
- *Duplo*
- *Senha*
- *Referência de objeto*
- *MapaDeValor*

Todos os tipos de dados podem ser configurados para conter um único valor ou vários valores indexados (chamados de *matriz*). Você pode definir um campo indexado como uma matriz de uma ou mais dimensões.


É possível editar objetos do conjunto de dados e conjuntos de dados do operador personalizado no Navegador da biblioteca. Você pode editar os conjuntos de dados para processos e cada operador no Criador de processos.



**Item: Descrição:**

- 1 **Conjuntos de dados no Navegador da biblioteca:** crie, edite e gerencie os objetos do seu próprio conjunto de dados no Navegador da biblioteca. Abra um conjunto de dados para editá-lo.
- 2 **Conjuntos de dados do operador personalizado:** crie, edite e gerencie seus próprios conjuntos de dados para objetos do operador personalizado no Navegador da biblioteca. Abra um operador personalizado para editar seu conjunto de dados.

---

Item:	Descrição:
3	<b>Conjuntos de dados no criador de processos:</b> a paleta Conjunto de dados exibe os conjuntos de dados disponíveis para o processo e cada operador. Você também pode consultar seus próprios conjuntos de dados no código. Clique no ícone  para abrir uma caixa de diálogo a fim de editar o processo ou o conjunto de dados do operador selecionado.
4	<b>Páginas, variáveis, tipos de dados e valores:</b> defina as páginas e as variáveis do conjunto de dados na metade esquerda da caixa de diálogo. Defina os tipos de dados e digite uma descrição na metade direita. Defina valores em cada lado. Clique com o botão direito do mouse no nome da variável de uma matriz para adicionar ou excluir um valor indexado.

---

## Criar um objeto de conjunto de dados nomeado

É possível criar e gerenciar objetos do conjunto de dados nomeado com os outros objetos de automação no Navegador da biblioteca.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. No painel de pastas do Navegador da biblioteca, selecione uma pasta.
3. Na barra de ferramentas, clique em Novo e selecione Conjunto de dados.  
Um novo objeto do conjunto de dados é exibido e está reservado para você.
4. Digite um nome para o novo conjunto de dados.
5. Clique duas vezes no conjunto de dados para definir as páginas e os campos.  
A caixa de diálogo Conjunto de dados é aberta.

## Tipos de conjuntos de dados

As *variáveis* do conjunto de dados (também chamadas de *campos*) podem conter valores literais que você define explicitamente no objeto do conjunto de dados. Você também pode atribuir valores às variáveis (campos) em tempo de execução usando expressões. Faça referência aos objetos do conjunto de dados e suas variáveis por nome usando expressões JavaScript em parâmetros calculados.

Você pode criar e configurar as variáveis do conjunto de dados para todos os tipos de conjuntos de dados, exceto o conjunto de dados do sistema. Faça referência às variáveis do conjunto de dados do sistema diretamente. É possível fazer referência aos conjuntos de dados do processo e do operador por meio do processo pai ou de objetos do operador. A tabela a seguir descreve os tipos de conjunto de dados.

Tipo de conjunto de dados:	Descrição e escopo:	Para referência em uma expressão:
Conjunto de dados nomeado	Os objetos do conjunto de dados armazenam a definição de um conjunto de dados nomeado no Navegador da biblioteca. As variáveis do conjunto de dados nomeado podem ser acessadas por qualquer operador, processo ou programação na mesma biblioteca. É possível editar a versão atual de um conjunto de dados nomeado, expandindo a pasta de biblioteca e clicando duas vezes no objeto do conjunto de dados.	Consulte <a href="#">Especificar variáveis do conjunto de dados nomeado</a> (na página 258).
Conjunto de dados do processo	Os conjuntos de dados do processo contêm variáveis que você ou outro criador define. O CA Process Automation também pode definir variáveis do processo automaticamente quando uma instância de processo é iniciada. Os conjuntos de dados do processo são exibidos na paleta Conjunto de dados do Criador de processos.	Consulte <a href="#">Especificar variáveis do conjunto de dados do processo</a> (na página 259).
Conjunto de dados do operador	Um conjunto de dados do operador é incluído em cada instância de um operador adicionado a um processo ou objeto de programação. O conjunto de dados do operador pode conter parâmetros do operador, variáveis definidas pelo usuário e variáveis definidas pelo programa. Um conjunto de dados do operador é mais acessível ao operador imediato e, em segundo lugar, a outros operadores de um processo. Os conjuntos de dados do operador são exibidos abaixo dos conjuntos de dados do processo na paleta Conjunto de dados do Criador de processos.	Consulte <a href="#">Especificar variáveis do conjunto de dados do operador</a> (na página 261).
Conjunto de dados do sistema	Contém variáveis predefinidas que estão disponíveis no contexto de todo o domínio do CA Process Automation. Essas variáveis acessam os parâmetros do sistema e são disponibilizadas pela palavra-chave <i>Sistema</i> .	Consulte <a href="#">Especificar variáveis do conjunto de dados do sistema</a> (na página 262).

## Definir páginas e variáveis de conjuntos de dados

Defina as páginas, as variáveis e os valores em um conjunto de dados. Quando você executa um processo, os operadores podem fazer referência aos valores em conjuntos de dados.

Novos conjuntos de dados nomeados e conjuntos de dados do processo incluem uma página raiz padrão chamada *Parâmetros*. As páginas e variáveis de conjuntos de dados do operador variam de acordo com o operador. É possível criar novas variáveis (campos) ou editar as variáveis existentes. Você também pode adicionar páginas a variáveis do grupo de maneiras lógicas. Para editar uma variável ou seu valor, clique na variável ou no seu valor e faça as alterações. Você também pode alterar os valores nas variáveis (campos) do conjunto de dados de forma programática.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Selecione uma pasta e localize um conjunto de dados. Use os recursos de pesquisa opcionais, se necessário.
3. Clique duas vezes no conjunto de dados.  
A guia Conjunto de dados da caixa de diálogo Conjunto de dados é exibida.
4. Clique em Reservar.
5. Para criar uma página, clique em Adicionar página.
6. Para criar uma variável:
  - a. Selecione uma página.
  - b. Clique em Adicionar variável.
7. Para renomear uma página ou variável, clique duas vezes nela. Os nomes devem começar com uma letra e devem ter um tamanho máximo de 32 caracteres.
8. Para variáveis, selecione o tipo de dados e a página, digite um valor inicial e forneça uma descrição opcional.

### Página

Define a página que contém a variável. Para mover a variável para uma página diferente, selecione essa página. Para operadores personalizados, a página especificada aqui corresponde a um grupo de propriedades expansíveis na paleta Propriedades.

### Descrição

(Opcional) Fornece informações úteis sobre a variável quando ela é exibida posteriormente em um conjunto de dados. Se houver, uma dica de ferramenta exibirá o texto inserido aqui quando você mover o ponteiro do mouse sobre o nome ou o valor do campo.

### Valor

Especifica o valor padrão se o campo estiver em branco (desmarcado). Para campos do tipo número inteiro, longo e duplo, o valor padrão é 0. Para campos do tipo número inteiro, duplo, longo ou sequência de caracteres, é possível especificar seu próprio valor padrão aqui. Para exibir o conteúdo do campo Valor em uma janela separada, clique com o botão direito do mouse no campo Valor e selecione Expandir.


**Observação:** a opção Expandir está disponível somente para o tipo de dados de sequência de caracteres.

Você também pode inserir restrições para um campo destes tipos:

- Para campos do tipo *Booleano*, é possível selecionar True ou False.
- Para campos do tipo *Referência do objeto*, clique no botão de procura (...) para selecionar um objeto do conjunto de dados.
- Para campos do tipo *Data*, clique no botão de procura (...) para selecionar uma data.
- Os campos Objeto Java são somente leitura.

**Observação:** o tipo de campo MapaDeValor não pode receber um valor padrão.

9. Para definir matrizes:
  - a. Na guia Geral, marque a caixa de seleção Matriz e selecione *Único* ou *Duplo* no campo Dimensão da matriz.
  - b. Clique com o botão direito do mouse na variável e selecione Adicionar valor indexado.
  - c. Clique no campo Valor para inserir um valor.

**Observação:** para editar o conjunto de dados de um operador personalizado, abra-o, reserve-o e clique na guia Conjunto de dados. Também é possível editar conjuntos de dados do processo e do operador clicando em  na barra de título da paleta Conjunto de dados no Criador de processos.

## Tipos de dados variáveis

Os tipos de dados variáveis do conjunto de dados são mapeados para os tipos de valor do JavaScript.

Os tipos de dados diferentes para as variáveis são os seguintes:

### Booleano

Armazena e retorna verdadeiro ou falso.

Em expressões, esse tipo é mapeado para o tipo de dados booleano JavaScript.

**Referência de objeto**

Faz referência a qualquer tipo de objeto disponível no CA Process Automation, incluindo objetos disponíveis em uma biblioteca de automação, um touchpoint e grupos de touchpoints.

**Data**

Armazena e retorna uma data em um formato de data especificado nas propriedades do parâmetro.

Em expressões, esse tipo é mapeado para o objeto de data JavaScript.

**Duplo**

Armazena e retorna um valor decimal. Dois valores são inseridos no seguinte formato:

```
[digits][.digits][[E|e][+|-]digits]
```

Em expressões, este tipo é mapeado para o tipo literal ponto flutuante JavaScript. O literal possui um valor mínimo de -1.7976931348623157E308 e um valor máximo de 1.7976931348623157E308.

**Número inteiro**

Armazena e retorna um valor de número inteiro de 16 bits. Um campo de número inteiro pode retornar um único número inteiro ou uma lista indexada de números inteiros. O número inteiro permite representar todos os números inteiros a dez dígitos de -2,147,483,648 a + 2.147.483.647.

Em expressões, esse tipo de valor é mapeado para o tipo de número inteiro do JavaScript.

**Objeto Java**

Armazena um objeto Java.

**Número longo**

Armazena e retorna um número inteiro de 32 bits. O tipo de dados número longo permite representar todos os números inteiros a 19 dígitos de -9,223,372,036,854,775,808 a + 9,223,372,036,854,775,807.

Em expressões, esse tipo de valor é mapeado para o tipo de número inteiro do JavaScript.

**Senha**

Retorna uma senha em formato criptografado. Os valores do campo de senha só podem ser atribuídos a outros campos de senha. Usuários não autorizados são impedidos de visualizar as senhas em um formato não criptografado.

**Observação:** o CA Process Automation protege as senhas de qualquer modificação. Operações como concatenação e manipulação de sequências de caracteres nas senhas resulta em um valor nulo.

**Sequência de caracteres**

Retorna uma sequência de caracteres. Um campo de sequência de caracteres pode retornar uma única sequência de caracteres ou uma lista indexada de sequência de caracteres.

Em expressões, este tipo de valor é mapeado para o tipo de sequência de caracteres JavaScript.

**MapaDeValor**

Define uma estrutura de conjunto de dados que é armazenada em outro conjunto de dados. É possível adicionar páginas e variáveis a um campo ValueMap. As variáveis e páginas de ValueMap podem ser aninhadas uma dentro da outra.

Em campos calculados, os campos de tipo ValueMap são tratados hierarquicamente dentro de um caminho do conjunto de dados. O exemplo a seguir trata o parâmetro *param1* no campo *VMap2* de ValueMap que, por sua vez, está aninhado no campo *VMap1* de ValueMap que é integrante do conjunto de dados *MyDataset*.

```
Process.MyDataset.VMap1.VMap2.param1
```

**Importante:** Alterar o tipo de dados ou a opção da caixa de seleção Matriz elimina quaisquer dados existentes no campo. Edite o conjunto de dados somente se for aceitável descartar qualquer valor atual do campo.

**Configurações de validação para variáveis do conjunto de dados**

Quando você seleciona o tipo de variável, é possível especificar exatamente qual o tipo de dado um usuário pode inserir. As restrições que podem ser aplicadas a um valor dependem do tipo de dados selecionado. A tabela a seguir lista as possíveis restrições para os diferentes tipos de dados em um conjunto de dados do operador.

Tipo de dado	Validações
Booleano	Nenhuma
Data	Nenhuma
Duplo	Especifica os valores mínimos e máximos para uma variável tipo dupla entre menos 1.7976931348623157E308 e mais 1.7976931348623157E308. O valor mínimo é o valor mínimo permitido para valores duplos. Valor máximo é o valor máximo permitido para os valores duplos.
Número inteiro	Especifica os valores mínimos e máximos para uma variável tipo número inteiro entre menos 2147483648 e mais 2147483647. O valor mínimo é o valor mínimo permitido para valores inteiros. O valor máximo é o valor máximo permitido para valores inteiros.
JsonObject	Nenhuma

Tipo de dado	Validações
Número longo	Especifica os valores mínimos e máximos para uma variável tipo longo entre menos 9223372036854775808 e mais 9223372036854775807.  O valor mínimo é o valor mínimo permitido para valores de número longo.  O valor máximo é o valor máximo permitido para valores de número longo.
Objeto	Nenhuma
Senha	Nenhuma
Sequência de caracteres	Nenhuma
MapaDeValor	Nenhuma

## Matrizes com valores indexados

Você pode definir uma variável do conjunto de dados para armazenar um único valor ou uma matriz de valores indexados. Você pode acessar cada valor em uma matriz indexada. É possível definir os campos indexados em uma matriz de dimensão única ou dupla.

Cada dimensão em uma matriz começa com um valor indexado [0] e continua na sequência com [1], [2], [3], e pode continuar com qualquer quantidade de valores adicionais. Em expressões, represente uma dimensão ou um nível de uma matriz com notação de colchetes. Cada nível de uma matriz oferece suporte às propriedades e aos métodos da matriz JavaScript.

O tipo padrão de campo é para armazenar um único valor. Para especificar que um campo deve armazenar uma lista indexada de valores, marque a caixa de seleção **Matriz** e configure os valores indexados.

## Definir um MapaDeValor como uma matriz

Você pode criar um MapaDeValor, definir suas variáveis e incluir matrizes nele. Também é possível optar por definir um MapaDeValor como uma matriz. Cada variável de MapaDeValor representa uma estrutura que você define no conjunto de dados em *Tipo de elemento*.

1. Clique na guia Biblioteca.
2. Selecione uma pasta e localize um conjunto de dados. Use os recursos de pesquisa opcionais ou crie um conjunto de dados, se necessário.
3. Clique duas vezes no conjunto de dados.  
A guia Conjunto de dados da caixa de diálogo Conjunto de dados é exibida.
4. Clique em Reservar.
5. Selecione uma página. A página padrão é chamada *Parâmetros*.
6. Na barra de ferramentas, clique em Adicionar variável.
7. Na guia Geral:
  - a. Defina o tipo para MapaDeValor.
  - b. Não clique na caixa de seleção Matriz.
8. Expanda a nova variável de MapaDeValor e selecione uma página. A página padrão é chamada *Parâmetros*.
9. Na barra de ferramentas, clique em Adicionar variável.
10. Na guia Geral:
  - a. Configure o tipo.
  - b. Insira um valor inicial.
  - c. Marque a caixa de seleção Matriz.
11. Repita as últimas três etapas. Selecione uma página, clique em Adicionar variável e defina a variável como uma matriz.
12. Expanda a variável e página de MapaDeValor. Clique com o botão direito do mouse na primeira variável e selecione Adicionar valor indexado.
13. Repita a última etapa.
14. Clique com o botão direito do mouse na segunda variável e selecione Adicionar valor indexado.
15. Repita a última etapa.  
Você definiu um MapaDeValor constituído de duas matrizes.
16. Clique na variável de MapaDeValor original.
17. Na guia Geral, marque a caixa de seleção Matriz.  
A pasta Tipo de elemento é exibida sob a variável de MapaDeValor.

18. Expanda totalmente Tipo de elemento para exibir suas páginas, suas variáveis e seus valores indexados.
19. Clique com o botão direito do mouse na variável de MapaDeValor original e escolha Adicionar valor indexado.  
O índice de matriz [0] é exibido.
20. Expanda o índice de matriz [0]. As páginas, as variáveis e os valores indexados são copiados da estrutura Tipo de elemento.

Ao adicionar uma variável a uma página em Tipo de elemento, todas as entradas de índice da matriz de mapa de valor existentes incluem imediatamente a nova variável sob a página correspondente.

Quando você exclui uma variável de uma página em Tipo de elemento, todas as entradas de índice da matriz de mapa de valor existentes deixam de incluir a variável excluída sob a página correspondente.

Ao especificar valores para as variáveis em Tipo de elemento, eles se tornam valores padrão para qualquer nova entrada indexada da matriz de mapa de valor. Os novos valores de Tipo de elemento não são propagados para as entradas de índice existentes da matriz de mapa de valor.

## Modificar um conjunto de dados

Você pode modificar um conjunto de dados adicionando valores indexados ou modificando dados da variável. Esse procedimento fornece menos recursos do que as tarefas mais robustas de conjunto de dados de definição de páginas e variáveis.

### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse em um conjunto de dados e selecione Modificar conjunto de dados.
3. Na janela Modificar conjunto de dados:
  - a. Clique com o botão direito do mouse em uma variável da matriz e selecione Adicionar valor indexado para adicionar uma entrada à matriz.
  - b. Clique em um valor para adicioná-lo ou alterá-lo diretamente.

## Exibir ou copiar uma expressão do conjunto de dados

Use este procedimento para exibir ou copiar a referência completa a uma matriz de conjunto de dados ou a um valor específico.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Na Biblioteca, execute uma das seguintes ações:
  - Clique duas vezes em um conjunto de dados.
  - Clique com o botão direito do mouse em um conjunto de dados e clique em Editar.
  - Clique com o botão direito do mouse em um conjunto de dados e clique em Modificar conjunto de dados.
3. Na janela resultante, clique em uma matriz ou uma variável específica para exibi-la.
4. Clique com o botão direito do mouse na matriz ou variável e clique em Exibir expressão.

A janela Expressão de conjunto de dados é exibida.

5. Exiba ou copie a expressão. Também é possível arrastar a expressão para um campo de destino que aceite expressões e ofereça suporte ao recurso de arrastar e soltar.

## Ler valores do sistema operacional em variáveis do conjunto de dados

O CA Process Automation pode ler os valores gerados por um processo shell nas variáveis do conjunto de dados. Antes de um operador de script do UNIX ou de script do Windows executar seu script associado, ele cria uma pasta para aceitar valores gerados pelo script. A variável de ambiente C2OSVD especifica o caminho completo para a pasta (por exemplo, C:\TMP\\_VS\_559) criada para o operador de script. Um script pode, em seguida, copiar os dados para os arquivos de texto na pasta para passar os dados de volta para o CA Process Automation. Os dados transmitidos de volta para o CA Process Automation usando o diretório C2OSVD preenchem as variáveis no conjunto de dados do operador de script.

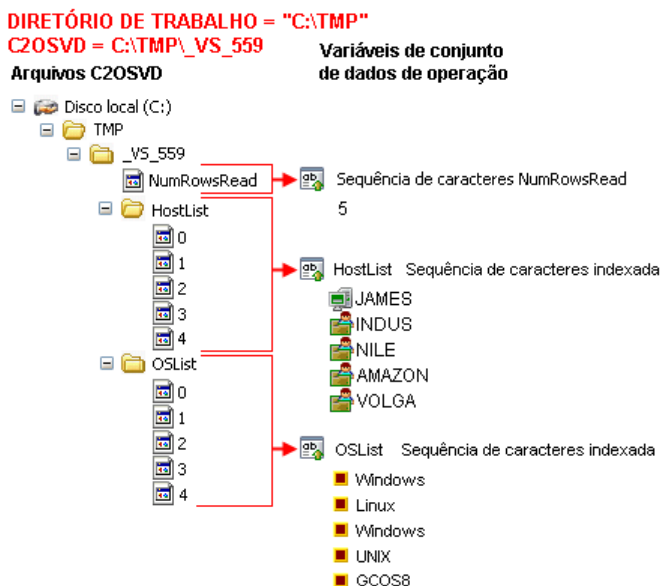
Um script deve salvar os dados em arquivos de texto no diretório C2OSVD. Após um operador de script concluir seu script (mas antes de executar as ações de execução posterior), ele determina se o local apontado pelo C2OSVD contém arquivos. O CA Process Automation então cria variáveis do conjunto de dados do operador de acordo com as seguintes regras:

- O operador salva o conteúdo de um arquivo do local C2OSVD para uma variável de sequência de caracteres no conjunto de dados do operador. A variável da sequência de caracteres tem o mesmo nome do arquivo.
- Uma pasta no local C2OSVD gera uma variável indexada no conjunto de dados do operador, com o mesmo nome da pasta. O operador salva o conteúdo dos arquivos de texto nomeados em uma sequência numerada localizada na pasta para os elementos correspondentes na variável indexada. Os elementos são numerados de 0 a quantos elementos como o número mais alto de arquivos na pasta. Arquivos ausentes na sequência numérica gera elementos nulos na variável indexada.
- O CA Process Automation requer um script para criar arquivos no local C2OSVD com os mesmos nomes das variáveis que você deseja exibir no conjunto de dados do operador. Se o script criar arquivos de texto com extensões de arquivo (como .txt), o CA Process Automation incluirá a extensão no nome da variável.

A ilustração a seguir mostra a conversão de arquivo para variável quando o diretório de trabalho é definido como C:\TMP. O operador anexa \\_VS\_599 ao caminho do diretório de trabalho para criar o valor C:\TMP\\_VS\_599 do C2OSVD. O nome da pasta (neste caso \_VS\_599) é exclusivo para cada instância de qualquer operador.

A ilustração também mostra duas pastas %C2OSVD%\HostList e %C2OSVD%\OSList que contêm cinco arquivos chamados 0, 1, 2, 3 e 4. O script grava um nome de computador em cada arquivo numerado na pasta HostList. O script grava um nome de sistema operacional em cada arquivo numerado na pasta OSList. O CA Process Automation cria duas variáveis indexadas com base zero após executar o script, HostList e OSList. O aplicativo atribui o conteúdo dos arquivos numerados nas pastas HostList e OSList para os elementos correspondentes das variáveis indexadas.

No exemplo, o script do Windows usa a variável C2OSVD para criar o arquivo %C2OSVD%\NumRowsRead. O CA Process Automation cria uma variável correspondente (NumFilesRead) no conjunto de dados do operador de script após executar o script. O produto então atribui o conteúdo do arquivo NumFilesRead à variável.



Quando um operador de script é concluído, ele exclui a pasta C2OSVD e seu conteúdo. O código de execução posterior do operador de script pode acessar as variáveis do conjunto de dados do operador. Geralmente, o código copia os valores da variável do conjunto de dados do operador para as variáveis locais no conjunto de dados do processo ou nas variáveis do conjunto de dados do operador em operadores do processo subsequentes. Os três scripts de exemplo deste capítulo mostram como o exemplo ilustrado nesta seção é implementado com o uso do script do UNIX, VBScript ou PerlScript.

## Os scripts de exemplo para ler valores do sistema operacional nas variáveis do conjunto de dados

Um processo pode usar o script do UNIX, VBScript ou PerlScript para ler as variáveis do sistema operacional nas variáveis do conjunto de dados do operador. Cada script lê linhas de um arquivo de texto que especifica um nome de host, um único espaço e o sistema operacional em execução no host. Por exemplo:

```
JAMES Windows
INDUS Linux
NILE Windows
AMAZON UNIX
YANGTZE Solaris
GILA UNIX
```

Os três scripts salvam os nomes de host em um campo indexado chamado HostList e os nomes de sistema operacional em um campo indexado chamado OSList.

### O processo incorpora estas etapas:

1. O módulo do UNIX ou do Windows passa o nome de um arquivo de texto. O módulo obtém o nome do arquivo a partir de um parâmetro na guia Processo apropriada das propriedades do operador de script.
2. O módulo de UNIX ou Windows cria e inicializa a variável de ambiente C2OSVC com um caminho para uma única pasta.

Uma instância de um operador de script usa a pasta que a variável de ambiente C2OSVD especifica. A instância não é repetida nem substituída por qualquer instância subsequente desse ou de qualquer outro operador de script.

3. O módulo do UNIX ou do Windows cria a pasta referenciada pela variável de ambiente C2OSVD.
4. O script cria pastas chamadas HostList e OSList na pasta referenciada pela variável de ambiente C2OSVD.
5. O script lê cada linha do arquivo de texto e executa as seguintes ações:
  - O script grava os nomes de host em arquivos numeradas sequencialmente na pasta %C2OSVD%\HostList.
  - O script grava os nomes de sistema operacional em arquivos numerados sequencialmente na pasta %C2OSVD%\OSList.

Esses arquivos geram duas variáveis do conjunto de dados do operador indexado, HostList e OSList.

6. O script grava o número de linhas lidas para os arquivos em um arquivo chamado %C2OSVD%/NumRowsRead.

Esse arquivo gera uma variável de conjunto de dados do operador chamada NumRowsRead.

## Exemplo de script do UNIX: Operador de script do UNIXGetInfo

O operador UNIXGetInfo executa o script do UNIX em um touchpoint do UNIX.

### Script (UNIX)

O exemplo de script do UNIX cria dois campos indexados, *HostList* e *OSList*, no conjunto de dados do operador UnixGetInfo. Ele lê cada linha do arquivo de origem e atribui nomes de host (\$host) aos valores indexados nos campos *HostList* sequenciais e sistemas operacionais (\$opsys) aos valores indexados nos campos *OSList* numerados sequencialmente, começando com 0 e terminando em um menos o número de linhas lidas a partir do arquivo de origem.

As variáveis de parâmetro de script shell do UNIX \$1, \$2, etc., são definidas pela primeira, segunda, etc., entradas da área de entrada de parâmetros do operador de chamada *Executar o script*. Nesse caso, apenas um parâmetro está sendo transmitido, o qual é usado para definir SourceFile. O número de linhas lidas é atribuído à variável de conjunto de dados do operador denominada *NumRowsRead*. A linha `sleep 30` não tem nenhuma outra finalidade além de pausar o operador e dar ao usuário 30 segundos para ele examinar as pastas e os arquivos criados no local C2OSVD. Essa linha não seria incluída em um script de produção.

```
#!/bin/ksh
SourceFile=$1
Nº conhecido para ler 2 variáveis, host e sistema operacional
mkdir $C2OSVD/HostList
mkdir $C2OSVD/OSList
integer counter=0
ao ler o host opsys; fazer
    echo -n $host > $C2OSVD/HostList/${counter}
    echo -n $opsys > $C2OSVD/OSList/${counter}
    counter=$((counter+1))
done < $SourceFile
echo -n $counter > $C2OSVD/NumRowsRead
sleep 30
exit 0
```

## Exemplo VBScript: informações do script WinGetInfo

O operador de script WinGetInfo executa o VBScript em um touchpoint do Windows.

## Script (VBScript)

O script cria dois campos indexados, HostList e OSList, no conjunto de dados do operador WinGetInfo. O script:

- Lê o arquivo de origem, linha por linha.
- Atribui nomes de host (strHost) a valores indexados em campos HostList sequenciais. A numeração começa em 0 e termina em um número a menos que o número de linhas lidas no arquivo de origem.
- Atribui sistemas operacionais (strOS) aos valores indexados em campos OSList numerados sequencialmente. A numeração começa em 0 e termina em um número a menos que o número de linhas lidas no arquivo de origem.

O script preenche a variável oArgs.Item com as entradas da área de entrada de parâmetros da chamada do operador Executar o script. A primeira entrada preenche oArgs.Item(0). O script atribui o número de linhas que lê à variável do conjunto de dados do operador NumRowsRead. A linha Wscript.sleep 30000 pausa o operador por 30 segundos para que o usuário possa examinar as pastas e os arquivos criados no local C2OSVD. Você não incluiria a linha Wscript.sleep 30000 em um script de produção.

```
Esmecer oArgs
Esmecer oShell
Esmecer colProcessEnv
Esmecer objFSO
Esmecer objDir
Esmecer objFileIn
Esmecer objFileOut
Esmecer intCounter
Esmecer intExitCode
Esmecer strLine
Esmecer intSpacePos
Esmecer strHost
Esmecer strOS
no próximo resumo de erro
definir oArgs = WScript.Arguments
definir oShell = WScript.CreateObject ("WScript.Shell")
definir colProcessEnv = oShell.Environment ("Processo")
if oArgs.Count = 1 then          'must have the required argument
                                to proceed normally,
                                fails if more arguments are present
    strSourceFile = oArgs.Item(0)
    Definir objFSO = CreateObject("Scripting.FileSystemObject")
    Definir objDir = objFSO.CreateFolder(colProcessEnv("C20SVD"))
    Definir objDir = objFSO.CreateFolder(colProcessEnv("C20SVD") & "\HostList")
    Definir objDir = objFSO.CreateFolder(colProcessEnv("C20SVD") & "\OSList")
    intCounter = 0
    Definir objFileIn = objFSO.OpenTextFile(strSourceFile, 1)
    Fazer até objFileIn.AtEndOfStream
        strLine = objFileIn.ReadLine
        intSpacePos = InStr(strLine, " ")
        strHost = Left(strLine, intSpacePos - 1)
        strOS = Right(strLine, Len(strLine) - intSpacePos)
        Definir objFileOut = objFSO.CreateTextFile(colProcessEnv("C20SVD") &
            "\HostList\" & intCounter)
        objFileOut.Write strHost
        objFileOut.Close
        Definir objFileOut = objFSO.CreateTextFile(colProcessEnv("C20SVD") &
            "\OSList\" & intCounter)
        objFileOut.Write strOS
        objFileOut.Close
        intCounter = intCounter + 1
    Loop
    objFileIn.close
    Definir objFileOut = objFSO.CreateTextFile(colProcessEnv("C20SVD") &
        "\NumRowsRead")
    objFileOut.Write intCounter
    objFileOut.Close
    intExitCode = 0
else
```

```
        'WScript.echo "argumento incorreto ou argumento necessário NÃO está presente"  
        intExitCode = 5  
terminar se  
Wscript.sleep 30000  
on error goto 0  
Wscript.Quit intExitCode
```

### Exemplo PerlScript: operador de script WinGetInfoPerl

O operador WinGetInfoPerl executa o script PerlScript em um touchpoint do Windows. No exemplo de processo, PerlScript é executado em um touchpoint do Windows, embora também possa ser executado em um touchpoint do UNIX.

#### Script (PerlScript)

O script cria dois campos indexados, HostList e OSList, no conjunto de dados do operador WinGetInfoPerl. O script:

- Lê o arquivo de origem, linha por linha.
- Atribui nomes de host (\$host) a valores indexados em campos HostList sequenciais. A numeração começa em 0 e termina em um número a menos que o número de linhas lidas no arquivo de origem.
- Atribui sistemas operacionais (\$opsys) a valores indexados em campos OSList numerados sequencialmente. A numeração começa em 0 e termina em um número a menos que o número de linhas lidas no arquivo de origem.

O script preenche a variável ARGV com as entradas da área de entrada de parâmetros da chamada do operador Executar o script. O script atribui o número de linhas que lê à variável do conjunto de dados do operador NumRowsRead. A linha sleep 30 pausa o operador por 30 segundos para que o usuário possa examinar as pastas e os arquivos criados no local C2OSVD. Você não incluiria a linha sleep 30 em um script de produção.

```
use strict;
my $filename = "";
imprimir " script de exemplo para recuperar os dados no nível OS em variáveis dentro
do C20\ n\ n";
my $numargs = @ARGV;
if ($numargs == 1) {
    $filename = shift @ARGV;
} else {
    imprimir "digite o caminho e o nome do arquivo para processar:\n";
    chomp($filename = <STDIN>);
}
my $c2osvd = $ENV{'C2OSVD'};
mkdir $c2osvd;
mkdir $c2osvd . "/HostList";
mkdir $c2osvd . "/OSList";
open HANDLE, $filename or die "ERROR: unable to open $filename: !\n";
my $counter = 0;
while (<HANDLE>) {
    my @fields = split;
    my $host = $fields[0];
    my $opsys = $fields[1];
    my $filename1 = $c2osvd . "/HostList/" . $counter;
    open HH, "> $filename1" or die "ERROR: unable to open $filename1: !\n";
    print HH $host;
    close HH;
    my $filename2 = $c2osvd . "/OSList/" . $counter;
    open HH, "> $filename2" or die "ERROR: unable to open $filename2: !\n";
    print HH $opsys;
    close HH;
    $counter++;
}
my $filename3 = $c2osvd . "/NumRowsRead";
open HH, "> $filename3" or die "ERROR: unable to open $filename3: !\n";
print HH $counter;
close HH;
close HANDLE;
sleep 30;
```

## Parâmetros do processo

É possível definir parâmetros para operadores do processo. Os parâmetros podem aceitar tanto as sequências de caracteres literais ou as expressões. Você pode digitar um valor como uma sequência de caracteres sem qualquer tipo de delimitação. Parâmetros calculados aceitam valores como expressões JavaScript. Use aspas simples ou aspas duplas para delimitar sequências de caracteres literais no JavaScript.

**Observação:** para ajudá-lo a identificar os campos do CA Process Automation que não aceitam expressões, os rótulos são exibidos em itálico.

Parâmetros calculados permitem o seguinte:

- Manipulação do módulo de invocação de resultados e outras variáveis.
- Parametrização de operadores.
- Definições de condições de espera baseadas em expressões booleanas (pré-condições e condições de espera). Condições de espera podem ser usadas para atrasar o processamento e sincronizar o uso de recursos por sequências diferentes de operadores em execução simultaneamente.

É possível definir parâmetros usando variáveis de conjunto de dados. Variáveis de conjunto de dados estão disponíveis para processos nos seguintes contextos:

- Em um contexto do orquestrador do CA Process Automation, os conjuntos de dados são denominados *conjuntos de dados nomeados*. Os conjuntos de dados nomeados definem variáveis que são acessíveis a qualquer processo no mesmo orquestrador. Um conjunto de dados nomeado é acessado especificando-se seu nome de caminho completo em uma expressão. Para exibir ou editar um conjunto de dados nomeado, clique duas vezes no objeto de conjunto de dados no Navegador da biblioteca para abrir o objeto no Criador de conjuntos de dados.
- No contexto do processo, há o *conjunto de dados do processo*. O conjunto de dados do processo está disponível a qualquer operador em um processo. O conjunto de dados do processo é acessado especificando a palavra-chave processo em uma expressão. É possível editar as variáveis em um conjunto de dados do processo na paleta Conjunto de dados no Criador do processo.
- Para cada operador em um processo, há um *conjunto de dados do operador*. As variáveis em um conjunto de dados do operador estão disponíveis para o operador e para outros operadores no mesmo processo. Um conjunto de dados do operador é acessado especificando-se o nome do operador em uma expressão. É possível editar variáveis em um conjunto de dados do operador na paleta Conjunto de dados no Criador do processo.

## Propriedades do operador

Esta seção fornece informações sobre os tipos de propriedades do operador.

## Sequências de caracteres literais

Para usar um valor de sequência de caracteres literal em um campo que aceita uma expressão (seu rótulo não está em itálico), coloque a sequência de caracteres entre os delimitadores. Use aspas simples ou duplas. Por exemplo, é possível digitar uma sequência de caracteres literal que especifica o caminho para um programa iniciar um processo UNIX, como segue:

```
"/usr/smart/program"
```

## Caractere de escape em sequências de caracteres literais

É possível usar o caractere de escape (\) em sequências de caracteres literais. Em vez de ser analisado pelo intérprete de idioma do CA Process Automation, o caractere digitado após o caractere de escape será interpretado literalmente. Se uma ação de semântica for vinculada a um caractere de escape, o intérprete converterá a ação em seu caractere equivalente em vez de executar a ação de semântica.

Por exemplo, digamos que você deseja incluir um caractere de aspas duplas em uma sequência de caracteres delimitada por aspas duplas. Preceda o caractere de aspas com o caractere de escape, de modo que o analisador não o interprete como o delimitador da sequência de caracteres:

```
\"
```

Para incluir o caractere de barra invertida em uma sequência de caracteres, preceda-o com o caractere de escape:

```
\\
```

### Mais informações:

[Tipo de dados de sequência de caracteres](#) (na página 243)

## Especificar caminhos em sequências de caracteres literais

Quando você usa a nomenclatura de arquivos do Microsoft Windows em uma sequência de caracteres literal para especificar um caminho em uma expressão, barras invertidas devem ser seguidas por caracteres de escape, como segue:

```
"C:\\IT PAM\\import\\script_oral.bat"
```

Na maioria dos casos, use nomes de arquivo *normalizados*, com barras (/), mesmo quando especificar um caminho em um computador Microsoft Windows. Por exemplo:

```
"C:/IT PAM/import/script_oral.bat"
```

Quando você especificar um caminho para uma pasta ou um objeto em uma biblioteca, a pasta raiz será representada por uma barra inicial, seguida por uma hierarquia de pastas delimitada por barras. Por exemplo:

```
"/Production/Processes/failover_process"
```

## Variáveis do conjunto de dados em parâmetros

Além das seqüências de caracteres literais, é possível usar as variáveis de conjunto de dados em uma expressão calculada. Variáveis em uma expressão não são incluídas entre aspas. O nome do programa na seguinte expressão calculada inclui as variáveis e o texto literal:

```
"/usr/bin" + Datasets["/Application1/Settings"].ProgramName
```

Seleção de uma variável do conjunto de dados para salvar o corpo da resposta em uma variável de conjunto de dados.

Seleção de uma variável do conjunto de dados para salvar o corpo da resposta em uma variável de conjunto de dados.

### Item: Descrição:

- 1 **Variáveis e valores do conjunto de dados:** passe o mouse sobre o valor de uma variável do conjunto de dados para exibir a descrição de uma dica de ferramenta sobre a variável, se houver uma disponível.
- 2 **Variáveis de referência de conjuntos de dados no código de execução anterior:** a chave em expressões que fazem referência a variáveis do conjunto de dados.

Quando você desejar usar a saída de um operador como a entrada para outro operador, use a mesma referência de variável por nome. Todos os parâmetros de entrada são automaticamente convertidos nas variáveis do conjunto de dados de saída após a conclusão do processo.

## Caminhos relativos para conjuntos de dados

O CA Process Automation pode usar caminhos absolutos ou relativos ao acessar conjuntos de dados nomeados. Caminhos absolutos também são conhecidos como caminhos completos ou fixos.

### Exemplo 1

*Folder1* está sob a pasta raiz na biblioteca. *Folder1* contém dois objetos: *Process1* e *Dataset1*. Para abrir o *Process1*, clique duas vezes no operador Iniciar processo e localize o campo Nome do processo na paleta Propriedades.

Em vez de inserir um valor, você quer usar ou fazer referência ao valor em um campo que é chamado *ProcessName* em *Dataset1*. Para o caminho absoluto, você especificaria:

```
Conjuntos de dados ["/Folder1/Dataset1"].ProcessName
```

Também é possível especificar o caminho de *Dataset1* em relação ao *Process1*. A mesma expressão usando um caminho relativo é:

```
Conjuntos de dados["Dataset1"].ProcessName
```

Enquanto o CA Process Automation avalia a expressão do caminho relativo, ele procura o *Dataset1* na mesma pasta como *Process1*.

Se você mover *Dataset1*, o caminho absoluto não será mais válido. Para corrigir essa situação, você deve atualizá-lo. Entretanto, desde que estejam na mesma pasta, você pode mover *Process1* e *Dataset1* para qualquer lugar e o caminho relativo ainda será válido.

### Exemplo 2

Semelhante ao exemplo 1, você deseja usar um campo em um conjunto de dados. Desta vez, você deseja usar *Dataset2*, no nível raiz da biblioteca. Para o caminho absoluto, você especificaria:

```
Conjuntos de dados["/Dataset2"].ProcessName.
```

A mesma expressão usando um caminho relativo é:

```
Conjuntos de dados["../Dataset2"].ProcessName.
```

Essa expressão de caminho informa ao aplicativo para procurar na pasta qual é o pai de *Folder1* (a pasta que contém o processo). *Folder1* é o ponto de partida. O código `"../Dataset2"`, informa literalmente para passar *um nível para cima* na hierarquia de pasta e procurar por *Dataset2*. Nesse caso, a pasta pai de *Folder1* é a pasta raiz e é lá que o aplicativo procura por *Dataset2*.

Esses conceitos, resumidos nos dois pontos a seguir, também se aplicam ao Linux/UNIX, Windows e a qualquer ambiente que ofereça suporte a convenções de nomenclatura uniforme.

- Uma pasta pai "/" existe.
- Todas as outras pastas são filhos da pasta pai.

Quando essas condições existirem, você poderá simplificar expressões complexas usando caminhos relativos. Por exemplo:

`"../../"`

**Observação:** os caminhos relativos ou absolutos podem ser usados como expressões em qualquer objeto.

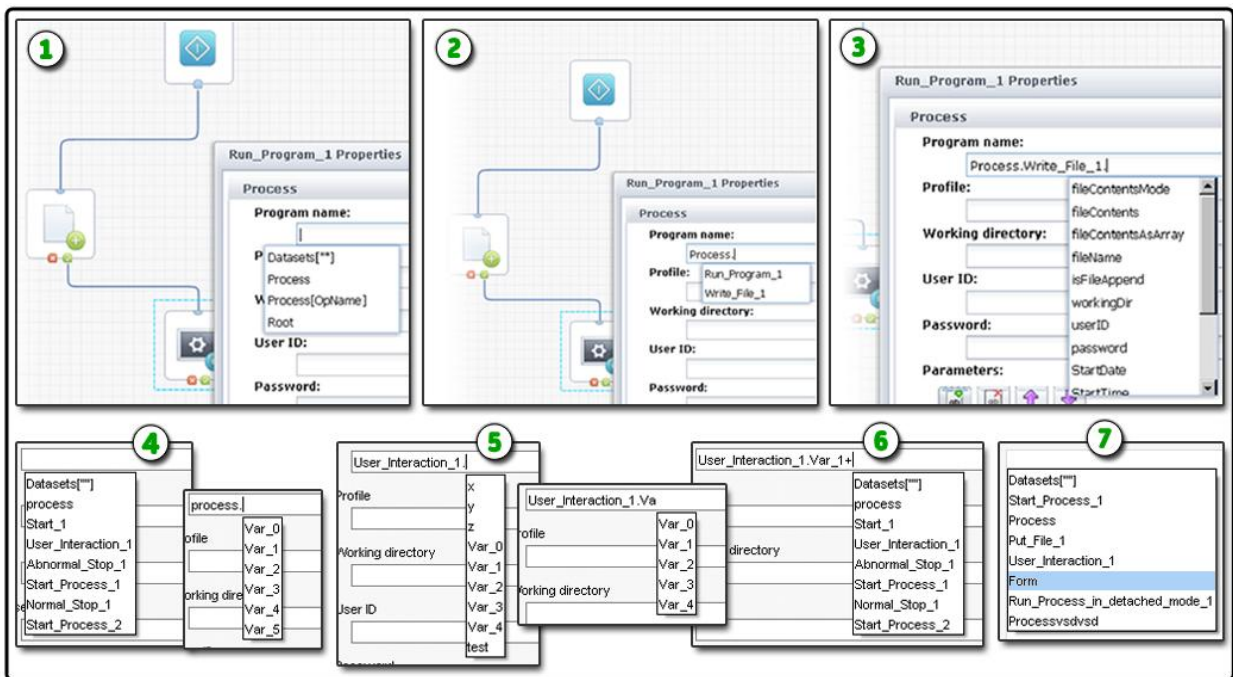
## Assistência de nome de variável do conjunto de dados

Os conjuntos de dados do CA Process Automation definem e armazenam os grupos de variáveis compartilhados entre as instâncias de processo. O CA Process Automation permite usar as variáveis como parâmetros de entrada na execução das instâncias do processo.

Após criar as variáveis de conjunto de dados dos processos e operadores, é possível que você não se lembre dos nomes de variável. As variáveis que fazem parte de um operador, e que não são algo que você definiu, são fáceis de serem esquecidas. Para ajudá-lo a fazer referência a essas variáveis de conjunto de dados do processo e de operador sem precisar se mover entre os operadores, o aplicativo fornece um assistente de edição em contexto. Conhecido como *Assistência do nome da variável do conjunto de dados* e chamado pressionando *Ctrl+Espaço*, esse recurso ajuda a:

- Identificar as variáveis do conjunto de dados e aplique-as a qualquer campo que aceite expressões.
- Reduzir o tempo de desenvolvimento do processo
- Reduzir a ocorrência de erros no escopo ou na sintaxe.

A maioria dos campos de texto que aceitam as expressões como entrada oferecem suporte à *Assistência do nome da variável do conjunto de dados*. Consulte a figura a seguir para obter exemplos.



Item:	Descrição:
-------	------------

Item:	Descrição:
1	<b>Valores sugeridos para o operador Executar programa:</b> neste exemplo, o usuário pressionou <b>CTRL+espaço</b> para exibir um pop-up da <i>Assistência do nome da variável do conjunto de dados</i> . Uma lista de valores nos níveis global e raiz é exibida.
2	<b>Operadores sugeridos:</b> após inserir Processo. uma lista de operadores aparece no nível de processo.
3	<b>Variáveis sugeridas:</b> após especificar o escopo do processo e o operador, uma lista de variáveis é exibida no nível de operador.
4	<b>Sugestões com base no escopo:</b> quando você insere informações no campo pela primeira vez, as sugestões são apropriadas nesse nível ou escopo. Depois que você especificar o parâmetro de processo, o escopo de sugestões será reduzido às variáveis apropriadas no nível de processo.
5	<b>Atualizações com base na entrada do usuário:</b> à medida que você digitar um texto, o aplicativo atualizará dinamicamente as sugestões. Neste exemplo, digitar Va reduz as opções possíveis somente às entradas correspondentes que começam com <i>Va</i> .
6	<b>Vários conjuntos de dados:</b> o aplicativo atualiza dinamicamente o escopo das sugestões quando você especifica expressões que abrangem mais de um conjunto de dados.
7	<b>Sugestões inteligentes:</b> quando apropriado, o aplicativo incluirá parâmetros adicionais com base no escopo e no contexto. Por exemplo, um operador de formulário pode incluir o parâmetro <i>Formulário</i> e um operador Iniciar processo pode incluir a palavra reservada <i>Chamador</i> . O aplicativo também oferece suporte automaticamente a um mapa de valores e a variáveis de matriz.

## Usar a Assistência de nome de variável do conjunto de dados

Use o recurso *Assistência do nome da variável do conjunto de dados* em vários contextos suportados em todo o aplicativo.

### Siga estas etapas:

1. Abra e reserve o objeto de automação a ser editado. Por exemplo, um processo, um formulário ou um conjunto de dados.
2. Pressione a tecla **Tab** ou clique em um campo de texto que aceite expressões.
3. Pressione **Ctrl + Espaço**.  
Uma lista de sugestões de valores é exibida.
4. Selecione o valor que deseja usar. Para selecionar variáveis de processo ou de conjunto de dados do operador, digite o seguinte valor de sequência de caracteres no campo de texto:

Processo.

5. Digite um nome de operador para a assistência do nome da variável específica do operador.
6. Para filtrar dados com base na entrada de texto, comece digitando ou inserindo caracteres.  
A lista de sugestões é atualizada dinamicamente à medida que você insere texto.
7. Selecione os valores que deseja usar.

## Parâmetros de senha

Os caracteres digitados no campo senha são mostrados como asteriscos (\*). As senhas salvas em um campo de conjunto de dados de tipo de senha são criptografadas. Uma expressão em um parâmetro calculado só pode atribuir o valor de um campo de senha para outro campo de senha.

## Configurações de execução

As *Configurações de execução* especificam como e onde executar um operador. As configurações de destino e de tempo limite estão disponíveis para operadores nos processos e programações. Os grupos Processamento e Loop só estão disponíveis para os operadores nos processos.

## Configurações de destino

O campo Destino na seção Configurações de execução da paleta Propriedades especifica onde o operador é executado. Use as diretrizes a seguir ao especificar um destino no campo Destino:

- Se o destino for um orquestrador, digite seu touchpoint. Você pode deixar o campo Destino em branco para que o destino seja o orquestrador que está executando o processo.  
**Observação:** não especifique o endereço IP de um computador que hospeda um orquestrador agrupado.
- Se o destino for um agente, digite seu touchpoint. Se o touchpoint for mapeado para vários agentes com a mesma prioridade, o destino de execução exato será selecionado para balanceamento de carga.
- Se o destino for um agente específico e o touchpoint estiver mapeado para vários agentes, digite sua ID de agente.
- Se o destino não tiver nenhum agente, mas tiver um touchpoint do proxy, digite seu touchpoint do proxy.
- Se o destino for um host remoto ao qual um grupo de hosts faz referência:
  - Selecione a opção do menu suspenso Fazer correspondência do destino apenas em grupos de hosts (Ativado).
  - Selecione a opção Pesquisar DNS ao fazer correspondência com o destino em grupos de hosts (Ativado) para identificar o DNS se estiver fazendo a correspondência do destino no grupo de hosts.
  - Selecione a opção O destino é uma expressão calculada. Crie um conjunto de dados que faz referência a um endereço IP ou FQDN de um host que atende aos critérios de padrão de um grupo de hosts.
  - Digite a referência de conjunto de dados para o endereço IP ou FQDN no campo Destino.

**Importante:** use um conjunto de dados para o endereço IP ou FQDN se o processo com esse operador estiver destinado a ser importado como pacote de conteúdo. Apenas um conjunto de dados pode ser modificado nesse cenário. Isto é, a entrada do campo Destino não pode ser modificada após a importação.

## Variáveis do conjunto de dados do operador

O conjunto de dados do operador contém variáveis associadas a um operador. É possível exibir, criar, editar ou excluir variáveis e seus valores associados no momento da criação. As variáveis estão disponíveis para o operador ao qual estão associados ou qualquer outro operador em seu processo assim que o processo for iniciado. Depois que o operador for executado, ele criará automaticamente outras variáveis no conjunto de dados do operador. Algumas variáveis são padrão e definem as informações como a hora de início, a hora de término e o resultado. Outras variáveis são específicas de cada operador.

É possível usar uma notação de ponto ou a notação de colchetes com expressões para acessar uma variável de conjunto de dados do operador de qualquer operador em um processo:

```
Process_name.Operator_name.field_name  
Process_name[OpName_expression].field_name  
Process_name[OpName_expression][field_name_expression]
```

**Observação:** você também pode usar `IconName` no lugar de `OpName`.

As expressões retornam o nome do operador ou da variável, conforme indicado. A sintaxe a seguir retorna um elemento em um campo indexado, em que *n* é o número de elementos:

```
field_name[n]
```

O código de execução anterior e posterior para um operador pode usar a palavra-chave `OpName` para acessar o nome do operador atual. Para especificar uma variável de conjunto de dados do operador no código de execução anterior ou posterior desse mesmo operador, use a seguinte sintaxe:

```
Process[OpName].field_name
```

Por exemplo, as seguintes instruções de código de execução posterior executam estas ações:

- Atribua o nome do operador e o valor da variável de resultado associada às variáveis do conjunto de dados do processo `iName` e `iResult`
- Crie uma variável do conjunto de dados do operador chamada `World`:

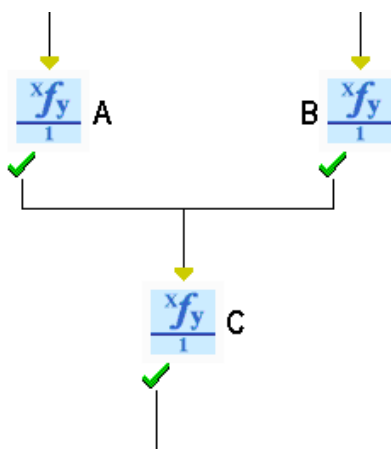
```
Process.iName = OpName  
Process.iResult = Process[OpName].Result  
Process[OpName].World = "Hello world!";
```

Se você souber o nome do operador no momento da criação, será possível usar o nome literal do operador para fazer referência às variáveis do conjunto de dados em uma expressão.

Ocasionalmente, no entanto, você não saberá o nome do operador em tempo de design. Por exemplo:

- Quando você estiver editando ações de execução anterior e posterior para um objeto de operador personalizado.
- Quando for necessário acessar as variáveis do conjunto de dados do operador em um dos vários operadores, mas você não souber qual operador foi executado até o tempo de execução.

É possível usar um código de execução anterior ou posterior em cada operador para salvar seu nome em uma variável, como fizemos com a variável `iName` no exemplo anterior. A variável poderia ser uma variável do conjunto de dados do processo ou poderia ser uma variável do conjunto de dados do operador pertencente ao operador que requer o nome. Por exemplo, na ilustração a seguir, o processo executa o OperadorA ou o OperadorB antes de atingir o OperadorC. O OperadorC pode então usar o nome da operação que o OperadorA ou o OperadorB salvou para uma variável de processo para acessar as variáveis do conjunto de dados.



Usar variáveis em vez de nomes fixos torna o código modular e intercambiável entre os operadores em um processo.

## Configurações das propriedades de processamento

As *propriedades de processamento* definem os códigos de execução anterior e posterior.

## Código de execução anterior e código de execução posterior

O código JavaScript de execução anterior e posterior é processado antes e após a execução de um operador. O código de execução anterior, em geral, é usado para configurar as variáveis de loop ou outras variáveis que podem ser usadas como parte do operador. O código de execução posterior, em geral, é usado para processar os resultados de um operador ou para aumentar os índices de loop.

A palavra-chave *OpName* pode ser usada para acessar o conjunto de dados do operador. Por exemplo, a instrução a seguir insere o nome do operador em uma mensagem e atribui a sequência de caracteres a uma nova variável de conjunto de dados do operador denominada *operatorMsg*:

```
Process[OpName].operatorMsg = "Recovery Operator" + OpName + "restructuring main  
server at "+System["Date"]+ ":" + System["Time"];
```

Normalmente, você deve incluir código que está estritamente associado ao processamento de seu operador específico.

Para códigos não relacionados, a prática recomendada consiste em adicionar um operador Cálculo separado ao processo.

### *Definir o status do operador*

Durante o processamento do código de execução anterior e posterior, você pode optar por especificar um valor para *setOperatorStatus*. Você pode forçar a falha ou o êxito do operador.

Para especificar o êxito do operador:

```
setOperatorStatus ("Success",Operation Result,"reason")
```

Para especificar a falha do operador:

```
setOperatorStatus ("Failure",Operation Result,"reason")
```

## Configurações de loop

A propriedade Loop especifica o número de vezes que um operador é repetido. Quando um operador é executado em um loop, as condições de saída e os links de conexão do operador são avaliados apenas quando o loop estiver concluído.

As configurações de loop tem as seguintes propriedades:

### Contagem de repetição

Especifica o número de vezes que um operador deve ser repetido. Esse valor pode ser especificado com um número inteiro ou uma expressão do CA Process Automation que retorna um número inteiro em tempo de execução. O valor padrão 1 executa um loop em um operador uma única vez em um fluxo de trabalho. Para executar um loop infinito, clique na caixa de seleção *Loop infinito*.

Uma expressão booleana também pode ser usada. A expressão é avaliada após o operador ser executado. Enquanto a expressão for avaliada como verdadeira, um operador em um fluxo de trabalho executará um loop contínuo. Se a expressão for falsa, o operador será encerrado.

### Loop infinito

Cria um loop infinito. O operador ou o processo é repetido continuamente até que o processo seja interrompido ou o loop seja parado a partir de uma ramificação diferente, usando um link de comando de interrupção de loop para o operador Loop.

### Atraso entre iterações

O CA Process Automation oferece suporte a uma opção de atraso inerente para cada operador que tenha uma opção de loop. O campo de texto *Atraso entre as iterações* usa uma expressão. A expressão é avaliada para um número inteiro e o valor é assumido como atraso em segundos. Antes da próxima iteração ser executada, há um atraso conforme especificado pelo usuário após uma iteração do loop.

O valor mínimo para o atraso é zero. O atraso padrão é zero segundos. O CA Process Automation considera o atraso de zero segundos para todas as entradas inválidas.

## Configurações de tempo limite

As configurações de tempo limite permitem que o usuário defina um tempo limite como parte de cada operador. Se o operador não for concluído no tempo especificado definido nas configurações de Tempo limite, a execução assumirá uma porta de saída de tempo limite. Os usuários ainda manterão a opção para finalizar a execução do operador e assumir o caminho do tempo limite ou deixar que o operador continue com a execução.

As configurações de tempo limite têm as seguintes propriedades:

### **Sem tempo limite**

Especifica que não há qualquer tempo limite definido para o operador (ativado por padrão).

Para especificar um valor de tempo limite, desmarque a caixa de seleção *Sem tempo limite*.

### **Tipo**

Especifica o tipo de tempo limite. Selecionar um dos seguintes tipos de tempo limite.

### **Duração**

Especifica a duração do tempo limite em segundos.

### **Data de destino**

Especifica a data (DD/MM/AAAA) e a hora (24 horas) do tempo limite.

### **Data e hora de duração/destino**

Define a duração do tempo limite ou a data de destino para o operador.

**Ação**

Selecione uma ação, entre as listadas abaixo, a ser executada mediante o término do tempo limite:

**Abandonar**

Especifica se o fluxo é abandonado após o tempo limite do operador. As seguintes ações serão executadas:

- A operação é executada no modo desvinculado.
- O operador irá expirar.
- É executado um Código de execução posterior.
- O fluxo do processo passa pela ramificação de tempo limite.
- O operador de atraso é executado.

**Anular**

Especifica se o fluxo é cancelado após o tempo limite do operador. As seguintes ações serão executadas:

- O operador é cancelado.
- É executado um Código de execução posterior.
- O fluxo do processo passa pela ramificação de tempo limite.
- O operador de atraso é executado.

**Continuar**

Especifica se o fluxo continua após o tempo limite do operador. As seguintes ações serão executadas:

- O operador e a operação estão em estado de execução.
- O fluxo do processo passa pela ramificação de tempo limite.
- O operador de atraso é executado.
- O código de execução posterior é executado após o operador ser executado.

**Redefinir**

Especifica se o fluxo é redefinido após o tempo limite do operador. As seguintes ações serão executadas:

- O operador e a operação estão em estado de execução e são redefinidos.
- É executado um Código de execução posterior.
- O fluxo do processo passa pela ramificação de tempo limite.
- O operador de atraso é executado.

## Parâmetros calculados

Os parâmetros nas caixas de diálogo e nas páginas de propriedades que aceitam expressões são chamados de *parâmetros calculados*. Os valores para os parâmetros calculados devem ser inseridos como expressões JavaScript. É possível usar expressões JavaScript para definir valores de conjuntos de dados, efetuar cálculos nos operadores de serviço do intérprete, como parte do código de execução anterior e posterior, e para especificar parâmetros sempre que uma expressão for permitida. A maioria dos campos aceita parâmetros calculados. Campos que não aceitam expressões como entrada têm rótulos em itálico.

## Expressões

Uma *expressão* é qualquer instrução lógica que o aplicativo pode avaliar para retornar um valor. Ela pode incluir qualquer combinação dos seguintes tipos de dados:

- números inteiros (incluindo longo, duplo etc.)
- sequências de caracteres
- funções
- variáveis
- referências a outros operadores
- JavaScript
- datas e horas
- mapas de valores
- valores literais
- valores calculados
- palavras-chave lógicas *E*, *Ou* e *Not* ou símbolos (&&, ||, !)
- operadores de comparação (==, !=, <, >, <=, >=, <>)
- parênteses delimitadores

Expressões são entradas válidas para todos os campos, incluindo operadores JavaScript, funções, portas de saída personalizadas e campos de propriedade do operador, exceto para os campos rotulados em itálico.

## Palavras reservadas em expressões

Um número de palavras são reservadas nas expressões do CA Process Automation. Esses incluem as palavras reservadas do CA Process Automation, as funções do sistema e as palavras-chave de JavaScript. Não use essas palavras como identificadores (como para a variável ou outros nomes de objeto) em expressões.

### Palavras reservadas do CA Process Automation

- Chamador
- DateAdjust
- Processo
- CurrentIndex
- FreeRes
- Tamanho
- Conjuntos de dados
- OpName
- Sistema

### Funções do sistema do CA Process Automation

Consulte o Guia de *Referência do Criador de Conteúdo* para obter informações detalhadas sobre cada uma das funções do sistema do CA Process Automation.

## Palavras-chave de JavaScript reservadas

- Break
- do
- if
- switch
- var
- case
- else
- em
- deste
- void
- capturar
- false
- instanceof
- throw
- while
- continue
- finalmente
- novo
- true
- com
- padrão
- para
- nulo
- try
- excluir
- função
- return
- typeof

## Tipos de dados

As expressões do CA Process Automation oferecem suporte para os tipos de dados JavaScript. Variáveis ou constantes representam dados.

## Tipo de dados booleanos

Valores booleanos têm dois valores possíveis: verdadeiro e falso. O JavaScript converte os literais verdadeiros e falsos para 1 e 0, quando necessário.

Valores booleanos são geralmente o resultado da comparação feita em expressões JavaScript. Valores booleanos são normalmente usados em estruturas de controle. Por exemplo, a declaração se-então JavaScript executa uma ação se é verdadeira e uma ação diferente se é falsa.

Os exemplos a seguir mostram todas as expressões booleanas válidas:

```
Process.A == 1
!(Process.A == 1)
(Process.A != 1)
(Process.A == 1) && (Process.B > 0)
```

## Tipo Date Data

O tipo de data armazena e retorna as datas das variáveis do conjunto de dados. O formato do tipo de data pode ser especificado como parte do tipo de dados. Por exemplo, é possível especificar que representa uma data como mês e dia ou dia e mês.

## Tipo de dados duplos

O tipo de dado numérico duplo pode ter um ponto decimal. A sintaxe de tradicional é usada para números reais. Um valor real é representado como parte do número, seguido por um ponto decimal e a Parte fracionário do número. Esse tipo pode armazenar números reais de -1,7976931348623157E308 a 1,7976931348623157E308.

Literais de ponto flutuante podem ser representados usando a observação exponencial — um número real seguido pela letra e (ou E), seguido por um sinal de adição (+) ou de subtração (-), seguido por um expoente integral no seguinte formato:

```
[digits][.digits][{E|e}{+|-}]digits
```

### Exemplos

2,718

2345.789

7.748E-5

## Tipo de dados de número inteiro

O tipo de dados de número inteiro de 16 bits pode ser digitado como valores literais em uma expressão. Você pode representar exatamente todos os números inteiros entre -2.147.483.647 a + 2.147.483.647.

## Tipo JavaObject Data

Esse tipo de dados permite armazenar objetos Java no CA Process Automation.

Todas as variáveis JavaObject são somente leitura. Seus campos CurrentValue e somente leitura estão desativados. Só é possível editar os campos a seguir:

- Tipo
- Página
- Descrição
- Matriz

Um JavaObject que não esteja em branco mostra seu tipo de classe no campo CurrentValue associado. O campo CurrentValue de um JavaObject vazio é definido como [JavaObject].

O valor real da variável JavaObject é a versão da sequência de caracteres serializada do objeto Java, mas o CA Process Automation não mostra essa sequência de caracteres serializada. Em vez disso, ele mostra o tipo de classe Java do objeto.

Um JavaObject criado manualmente está sempre vazio porque você não pode inserir seu valor diretamente no CA Process Automation. As variáveis JavaObject normalmente são salvas em um conjunto de dados após o término da execução do operador Executar código Java.

## Tipo de dados de número longo

O tipo de dados de número longo é um campo de 32 bits que podem ser digitado como valores literais em uma expressão. Você pode representar exatamente todos os números inteiros entre -9.223.372.036.854.775,808 a +9.223.372.036.854.775.808.

## Tipo de dados de senha

O tipo de dados de senha armazena senhas em um formato criptografado nas variáveis do conjunto de dados. O valor armazenado em uma variável de tipo de senha não é exibido pelos usuários.

## Tipo de dados de referência ao objeto

O tipo de dados de referência ao objeto armazena um caminho completo ou de referência para um objeto em uma biblioteca. É possível usá-lo em qualquer lugar que uma expressão requer o caminho para um objeto. O objeto Filtro de referência ao objeto no tipo de referência restringe-a para referenciar um ou mais tipos de objetos especificados.

## Tipo de dados de sequência de caracteres

É possível digitar as sequências de caracteres como valores literais em uma expressão. Funções e operadores de JavaScript em uma expressão retornam sequências de caracteres. Para distinguir entre os identificadores (como nomes de variáveis) e as sequências de caracteres literais, coloque as sequências de caracteres literais entre delimitadores de sequência de caracteres. É possível usar aspas simples ou duplas como delimitadores de sequência de caracteres. Por exemplo, para usar uma sequência de caracteres literal no nome de um programa para iniciar um processo do UNIX, digite como a seguir:

```
"/usr/smart/program"
```

Nas caixas de texto marcadas com um asterisco (\*), digite uma expressão ou uma sequência de caracteres literal entre aspas. O asterisco indica que o CA Process Automation avalia o conteúdo da caixa de texto como uma expressão. Não use aspas para delimitar expressões.

Caixas de texto em que se espera valores literais não são marcadas com um asterisco (\*). Em tais casos, não delimite as sequências de caracteres entre aspas.

O JavaScript usa o caractere de barra invertida (\) para escapar de sequências de caracteres literais. Se houver uma ação de semântica vinculada a um caractere de escape, o caractere será convertido para o seu caractere equivalente, em vez de executar a ação de semântica. Por exemplo, 'C:\\pam' é convertido para 'C:\pam'.

O CA Process Automation interpreta qualquer caractere que segue o caractere de escape literalmente, em vez de analisá-lo por meio do interpretador de idioma. Por exemplo, para incluir aspas duplas em uma sequência de caracteres, anteceda-o com o caractere de escape, \". Nesse caso, o analisador não interpreta as aspas duplas como o delimitador de sequência de caracteres. Para incluir a barra invertida em uma sequência de caracteres, preceda-a com o caractere de escape, \\.

Para incluir o mesmo caractere que você usa como delimitador de sequência de caracteres, escape o caractere quando não o usar como um delimitador. Por exemplo, a seguinte sequência de caracteres é delimitada por aspas simples. A sequência de caracteres inclui aspas simples como caracteres de escape e aspas duplas como caracteres literais:

O mecanismo de banco de dados não pode localizar banco de dados `'CHECKERS'` no servidor `"GAMES"`

A mesma sequência de caracteres, delimitada por aspas duplas, inclui as aspas simples como caracteres literais e as aspas duplas como caracteres de escape:

O mecanismo de banco de dados não pode localizar o banco de dados `'CHECKERS'` no servidor `\ "GAMES\"`

A tabela a seguir mostra as sequências de escape no JavaScript:

Sequência	Caractere representado
<code>\0</code>	O caractere NULO ( <code>\u0000</code> )
<code>\b</code>	Backspace ( <code>\u0008</code> )
<code>\t</code>	Guia horizontal ( <code>\u0009</code> )
<code>\n</code>	Nova linha ( <code>\u000A</code> )
<code>\v</code>	Guia vertical ( <code>\u000B</code> )
<code>\f</code>	Formulário feed ( <code>\u000C</code> )
<code>\</code>	Retorno de carro ( <code>\u000D</code> )
<code>\"</code>	Aspas duplas ( <code>\0022</code> )
<code>\'</code>	Apóstrofo ou aspas simples ( <code>\0u0027</code> )
<code>\\</code>	O caractere Latin-1 especificado por dois dígitos hexadecimais NN

## Tipo ValueMap Data

O tipo de dados ValueMap contém um conjunto de variáveis de vários tipos de dados. É possível usá-lo para criar um grupo de variáveis em um conjunto de dados. Esse tipo é também conhecido como um registro ou uma estrutura.

## Operadores de JavaScript

É possível usar os operadores de JavaScript para criar uma sequência de caracteres, número inteiro, booleano e expressões lógicas a partir de uma combinação de entidades (números inteiros, sequências de caracteres, funções e conjuntos de dados). O número de operandos que eles esperam que caracterize operadores de JavaScript. A maioria dos operadores de JavaScript é de operadores binários que combinam duas expressões em uma única expressão mais complexa.

JavaScript também oferece suporte a vários operadores unários que convertem uma expressão única em uma única expressão mais complexa. Esta seção abrange vários operadores de JavaScript que são mais comumente usados em expressões do CA Process Automation.

## Operadores de matriz e de acesso ao objeto

O JavaScript usa notação de ponto (.) para matrizes e acesso ao objeto. É possível acessar os elementos de uma matriz com notação de colchetes ([]) e os elementos de um objeto com notação de ponto (.). O JavaScript trata notação de ponto e de colchetes como operadores.

A notação de ponto usa o seguinte formato:

```
object.identifíer
```

O operando *identifíer* pode ser o nome literal da propriedade, do método ou do nome da variável (em um conjunto de dados) sem aspas simples ou duplas. O operando não pode ser uma sequência de caracteres ou uma variável que contenha uma sequência de caracteres.

A notação de colchetes usa os seguintes formatos:

```
array[expression] //
```

O operando *array* se refere a uma matriz e o operando [*expression*] é avaliado como um valor de número inteiro para um índice de matriz.

```
object [expression]
```

O operando *object* se refere a um objeto e o operando [*expression*] é avaliado como uma sequência de caracteres que nomeia uma propriedade do objeto.

**Observação:** ao contrário da notação de ponto, o segundo operando é um identificador, o operando [*expression*] é uma sequência de caracteres.

A notação de colchetes permite o acesso aos elementos da matriz e às propriedades do objeto. A notação de colchetes permite também o acesso às propriedades do objeto sem restringir o operando *identifíer* como faz a notação de ponto.

## Operadores de atribuição

O JavaScript fornece o operador de atribuição normal e operadores de atribuição aritmética que fornecem atalhos para operadores aritméticos comuns.

Operador	Exemplo:	Equivalente
=	a = b	
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	Um/=B	a = a / b
%=	a %= b	a = a % b

## Operadores aritméticos

O JavaScript usa os seguintes operadores para a combinação de valores de número inteiro:

Operador	Descrição
*	Multiplicação
/	Divisão
+	Adição ou mais unário
-	Subtração ou menos unário
%	Módulo
++	Aumentar
--	Diminuir

Cálculos aritméticos em uma expressão seguem as regras algébricas:

- Quando uma expressão contiver vários operadores aritméticos, a multiplicação e a divisão serão calculadas primeiro e, em seguida, a subtração e a adição.
- Quando os operadores forem da mesma ordem, eles serão calculados da esquerda para a direita.
- É possível usar parênteses para alterar a ordem de precedência. Os cálculos dentro de parênteses serão avaliados primeiro. Se os parênteses são aninhados, o cálculo aninhado mais profundamente tem precedência.

## Operador de concatenação de sequência de caracteres

O idioma interpretado tem o seguinte operador para a combinação de valores de sequência de caracteres.

Operador	Descrição
+ (sequência de caracteres)	Concatenação de sequência de caracteres

Usar o operador de sequência de caracteres para combinar ou concatenar duas ou mais sequências de caracteres em uma única sequência de caracteres. Por exemplo, a expressão "ABCD" + "123" retorna a sequência de caracteres concatenada "ABCD123".

## Operadores lógicos

O idioma interpretado usa os seguintes operadores lógicos (ou booleanos) para combinar os resultados das funções ou dos operadores booleanos.

Operador	Descrição
&&	Lógico E
	Lógico OU
!	Lógico NÃO

Os operadores lógicos retornam True ou False. Eles reconhecem nulo, 0, "" ou indefinido como False e qualquer outro operando diferente de zero como True.

## Operadores de igualdade e comparação

Os operadores de comparação são usados com sequências de caracteres e dados numéricos. Operadores de comparação avaliados para um valor booleano. Eles retornam verdadeiro ou falso de acordo com o resultado da condição de teste.

Operador	Descrição
==	Igual a
===	Identidade
!=	Diferente de
!==	Sem identidade
<	Menor que
<=	Menor que ou igual a
>	Maior do que

Operador	Descrição
>=	Maior que ou igual a
?:	Operador condicional terciário A expressão X1? X2:x3 Retorna x2 quando x1 é verdadeiro ou x3 quando x1 é falso.

## Operador de prioridade

O operador do idioma interpretado do CA Process Automation segue as regras de prioridade computacionais padrões, conforme mostrado na tabela a seguir. Os operadores no mesmo nível de prioridade são executados da esquerda para a direita.

Prioridade	Operador
1	. ( ) [ ]
2	++ -- - (unário) + (unário) !
3	* / %
4	+ (adição) - (subtração) + (concatenação da sequência de caracteres)
5	< <= > >=
6	=====
7	&&
8	
9	?:
10	= *= /= %= += -=

## Palavras-chave para acessar os conjuntos de dados

A seguinte tabela descreve as palavras-chave usadas para fazer referência aos conjuntos de dados em vários contextos:

Conjunto de dados	Contexto do conjunto de dados	Descrição
Conjuntos de dados	Conjunto de dados nomeado	<p>Use o seguinte formato para acessar um conjunto de dados nomeado em uma biblioteca do CA Process Automation:</p> <pre>Datasets[<i>dataset_path</i>].field_name</pre> <p><b>dataset_path</b></p> <p>Uma expressão do CA Process Automation que é avaliada como o caminho completo de um conjunto de dados nomeado na biblioteca atual.</p> <p>Por exemplo, o seguinte caminho faz referência a um conjunto de dados nomeado CxLinuxDev localizado na subpasta Data da pasta Demo da biblioteca:</p> <pre>Conjuntos de dados ["/Demo/Data/"]</pre>

Conjunto de dados	Contexto do conjunto de dados	Descrição
Processo	Conjunto de dados de um processo	<p>Acessa um conjunto de dados do processo no seguinte formato:</p> <p><code>Process.field_name</code></p> <p>ou</p> <p><code>Process[expression]</code></p> <p><code>Process.field_name</code></p> <p>ou</p> <p><code>Process[expression]</code></p> <p><b>field_name</b></p> <p>A variável do conjunto de dados.</p> <p><b>expression</b></p> <p>Uma variável ou outra expressão que retorna o nome de um campo. Por exemplo:</p> <p><code>Process.x = 5;</code></p> <p><code>Process.fn = "x";</code></p> <p><code>Process.y = Process[Process.fn];</code></p> <p>Um conjunto de dados do processo é definido em um processo. Sempre que um processo é iniciado, ele cria uma cópia de si mesmo (chamada de <i>instância</i> do processo), que inclui o conjunto de dados do processo. O objeto de processo original determina os valores iniciais para o conjunto de dados. Alterações em um conjunto de dados em uma instância do processo não afetam o original.</p>

Conjunto de dados	Contexto do conjunto de dados	Descrição
Chamador	Conjunto de dados do processo para um processo pai quando ele está iniciando um processo filho	<p>Passa valores entre processos em uma hierarquia de chamada quando um processo que usa o Operador Iniciar processo (em anexo, separado ou embutido) ou o operador Executar processo desvinculado que o Módulo de processos oferece suporte ao iniciar outro processo. A opção de código de inicialização do conjunto de dados do processo do operador Iniciar processo especifica essas atribuições.</p> <p>Por exemplo, quando o ProcessoA que está chamando o ProcessoB precisa inicializar os campos no conjunto de dados do ProcessoB, o ProcessoA especifica o ProcessoB como parte do código de inicialização do conjunto de dados do processo. Nesse contexto, o chamador se refere ao conjunto de dados do pai (ProcessoA) e o Processo se refere ao conjunto de dados do filho (ProcessoB).</p> <p>No contexto do código de inicialização do conjunto de dados do processo, a palavra-chave Processo é sempre necessária para fazer referência a uma variável no conjunto de dados do processo filho.</p> <p>Se você omitir as palavras-chave Processo e Chamador em um nome de variável no script do código de inicialização do conjunto de dados do processo, o CA Process Automation procurará apenas por uma variável com escopo de cálculo. Ele não verificará uma variável com nome semelhante no conjunto de dados pai ou filho. Por exemplo, o seguinte código falha se nenhum X de escopo de cálculo foi criado anteriormente no contexto de cálculo:</p> <pre>Process.X = Caller.X; Process.Y = X + 100;</pre>

Conjunto de dados	Contexto do conjunto de dados	Descrição
<i>none</i>	O cálculo atual e, em alguns casos, o conjunto de dados do processo	<p>Se você omitir a palavra-chave <code>Processo</code> no lado esquerdo de uma instrução de atribuição, o CA Process Automation sempre criará ou atribuirá um valor dentro do escopo do cálculo atual (uma variável de cálculo). Uma variável de cálculo existe enquanto o CA Process Automation está processando um campo de cálculo. Por exemplo, o código a seguir cria uma variável de cálculo igual ao valor 5:</p> <pre>a = 5</pre> <p>Se você omitir a palavra-chave <code>Processo</code> em outros contextos (como no lado direito de uma instrução de atribuição), o CA Process Automation procurará primeiro por uma variável de escopo de cálculo ou uma variável do processo com o mesmo nome.</p> <p>Por exemplo:</p> <pre>Process.a = 1 Process.b = 2 a = 5 x = a y = b</pre> <p>O CA Process Automation cria duas variáveis no conjunto de dados do processo (<code>a=1</code> e <code>b=2</code>) e duas variáveis de cálculo (<code>x=5</code>, <code>y=2</code>).</p>
<code>Process.operator_name</code> ou <code>Process[expression]</code>	Conjunto de dados do operador em um processo	<p>Permite o acesso a um conjunto de dados do operador, em que <i>operator_name</i> é uma sequência de caracteres que especifica o nome de um operador em um processo. Por exemplo:</p> <pre>Process.y = Process.emailOp.subject</pre> <p>A expressão retorna o nome de um operador em um processo, por exemplo:</p> <pre>Process.opName = "emailOp"</pre> <pre>Process.y = Process[Process.opName].subject</pre> <p><b>Observações:</b></p> <p>Para obter informações sobre como especificar os campos do conjunto de dados do operador e obter uma lista de campos gerados pelo sistema nos conjuntos de dados do operador, consulte <a href="#">Especificar as variáveis do conjunto de dados do operador</a> (na página 261).</p> <p>- Para obter mais informações sobre como especificar variáveis de operador nas configurações de propriedades do operador, consulte <a href="#">Variáveis do conjunto de dados em parâmetros</a> (na página 225).</p>

Conjunto de dados	Contexto do conjunto de dados	Descrição
Sistema	Conjunto de dados do sistema	<p>Permite o acesso ao conjunto de dados do sistema.</p> <p>Campos no conjunto de dados do sistema representam as informações do sistema do CA Process Automation, como o nome do host, a data e a hora. O conjunto de dados do sistema é somente leitura.</p> <p><b>Observação:</b> para obter mais informações sobre campos no conjunto de dados do sistema, como especificar variáveis do sistema e obter uma lista de variáveis do sistema, consulte <a href="#">Especificar as variáveis do conjunto de dados do sistema</a> (na página 262).</p>
Raiz	Conjunto de dados pai	<p>Ativa um processo filho embutido para acessar o conjunto de dados do processo de uma instância pai.</p> <p>Você pode usar a palavra-chave Raiz para acessar o conjunto de dados do processo da instância raiz.</p> <p><b>Exemplos:</b></p> <ul style="list-style-type: none"> <li>- ProcessoA inicia um processo embutido, ProcessoB. ProcessoB inicia outro processo embutido, ProcessoC. ProcessoC usa a palavra-chave Raiz para acessar o conjunto de dados de ProcessoA.</li> <li>- ProcessoA inicia o ProcessoB em um modo não embutido e o ProcessoB inicia o ProcessoC em modo embutido. ProcessoC tem acesso somente ao conjunto de dados do ProcessoB (e não ao ProcessoA) usando a palavra-chave Raiz.</li> </ul>

### Acessar os campos do conjunto de dados em expressões

Os valores de campo do conjunto de dados em uma expressão usam a notação de colchetes (*[expression]*) ou de ponto (*.field*) (ou ambas) para especificar um conjunto de dados ou um campo em um conjunto de dados.

## Sintaxe para especificar o valor de um campo

Um valor em um único campo não indexado é acessado em uma expressão usando a notação de ponto:

```
dataset_reference.field_name
```

O parâmetro *dataset\_reference* corresponde a uma das palavras-chave descritas anteriormente. O parâmetro *field\_name* corresponde a um nome de campo no conjunto de dados. O exemplo a seguir retorna o valor do campo Y em um conjunto de dados do processo:

```
Process.Y
```

Como alternativa, um valor pode ser acessado usando a notação de colchetes:

```
dataset_reference[field_name_expression]
```

O parâmetro *field\_name\_expression* é uma expressão que retorna o nome de um campo no conjunto de dados de referência. O exemplo a seguir retorna o valor do campo Y em um conjunto de dados do processo:

```
Process["Y"]
```

A prática recomendada é normalmente usar a notação de colchetes para uma expressão para especificar um conjunto de dados ou um nome de campo em um conjunto de dados.

## Especifique o valor de um elemento em um campo indexado

Para acessar um valor em um campo indexado (array), use a seguinte sintaxe:

```
dataset_reference.indexed_field_name[index]
```

O parâmetro *indexed\_field\_name* define o nome do campo no conjunto de dados. *Index* é um número inteiro que especifica um elemento indexado na matriz.

Para especificar a sequência de caracteres *indexed\_field\_name* com a notação de colchetes, use a sintaxe a seguir. No exemplo, *indexed\_field\_name\_expression* é uma expressão do CA Process Automation que retorna a sequência de caracteres *indexed\_field\_name*.

```
dataset_reference[indexed_field_name_expression][index]
```

Como as matrizes do JavaScript, os campos indexados do CA Process Automation são matrizes de matrizes em vez de matrizes verdadeiras. Para acessar um elemento em uma matriz de matrizes, use o operador [ ] duas vezes. Por exemplo, para os números inteiros chamados pelo campo de número inteiro indexado bidimensional, cada número inteiro[x] do elemento de lista é uma lista indexada de números inteiros. Para acessar um determinado número inteiro no campo indexado, é possível gravar os números inteiros[x][y] da expressão. Em geral, para qualquer campo indexado de  $n$ -dimensões, use a seguinte sintaxe para acessar qualquer elemento de dados:

*dataset\_reference.indexed\_field\_name[index1][index2] ... [indexn]*

O parâmetro *index* é um número inteiro ou uma expressão que retorna um número inteiro. O parâmetro tem um valor de 0 (para o primeiro valor em uma lista indexada) até o comprimento da lista menos 1. A avaliação do índice é circular, portanto quando o valor de *index* exceder o comprimento de uma lista indexada, a seguinte fórmula determinará o seu valor:

*Actual-index = index % length-of-the-list;*

Em outras palavras, para  $n$  elementos em um campo indexado, você obtém os seguintes resultados:

- Um índice de  $n$  retorna o elemento 0
- Um índice de  $n+1$  retorna o elemento 1
- Um índice de  $n-1$  retorna o último elemento

A seguinte tabela exhibe os resultados do acesso de elementos de um campo indexado em um conjunto de dados do processo:

Expressão	Descrição
value = Process.X[2]	Refere-se ao terceiro elemento de um campo indexado X do conjunto de dados do processo.
value = Process.X[18]	Para um campo X indexado com um tamanho de 19, essa expressão se refere ao elemento 19 de X, o mesmo que Process.X[18].
value = Process.X[Process.Y + 2]	Uma expressão que calcula o índice.
value = Process.A[5][2]	Retorna o valor em uma matriz bidimensional. O terceiro elemento na sexta lista indexada que a matriz define resolve a matriz.

## Acessar o comprimento de um campo indexado

O CA Process Automation oferece suporte à propriedade de comprimento do JavaScript para matrizes e sua própria propriedade de tamanho para acessar o comprimento de um campo indexado. A propriedade de comprimento é somente leitura; a propriedade de tamanho permite alterar o número de elementos em um campo indexado. A propriedade de comprimento usa a notação de ponto ou de colchetes para retornar o número de elementos em um campo indexado:

```
dataset_reference.indexed_field_name.length
```

```
dataset_reference[indexed_field_name_expression].length
```

A propriedade de tamanho funciona da mesma maneira, usando a observação ponto ou colchetes:

```
dataset_reference.indexed_field_name.size
```

```
dataset_reference[indexed_field_name_expression].size
```

Como um campo indexado é uma matriz com base zero, as propriedades de tamanho e comprimento sempre retornam um a mais que o índice para o último elemento em um campo. Assim, quando um comprimento ou tamanho retornar  $n$ , um campo conterá 0, 1, ...,  $n-1$  elementos indexados e o índice do último elemento na matriz será  $n-1$ .

Atribuir um novo valor para a propriedade de tamanho aumenta ou trunca o número de elementos em um campo indexado. Diminuir o valor da propriedade de tamanho remove os elementos da extremidade superior de um campo indexado e exclui valores armazenados nos elementos excluídos. O seguinte código usa a propriedade de tamanho para aumentar o comprimento de uma matriz X por um elemento e, em seguida, atribui 25 ao novo elemento:

```
Process.X.size = Process.X.size + 1;  
Process.X[Process.X.size - 1] = 25;
```

Para uma matriz multidimensional, a propriedade de tamanho ou comprimento retorna o número de elementos em um endereço de matriz ao qual ela está anexada.

Por exemplo, para uma matriz bidimensional chamada matriz [a][b], a seguinte sintaxe retorna o tamanho da primeira dimensão de matriz, com os elementos contendo 0... size1-1:

```
size1 = matrix.length
```

A seguinte sintaxe retorna o tamanho da segunda dimensão de matriz, fornecendo primeiro o elemento 2 da dimensão, com b contendo os elementos 0... size2-1 quando a = 2. contendo os elementos 0... size1-1:

```
size2 = matrix[2].length
```

O exemplo a seguir ilustra como endereçar elementos de um campo indexado multidimensional por um loop através de todos os elementos em um campo indexado bidimensional (uma matriz de matrizes) na variável do conjunto de dados do processo chamada `matrix`. O código atribui o valor de cada elemento de um campo indexado unidimensional na variável do conjunto de dados do processo chamada `values`:

```
var i; j; k=0;
para (i=0; i < Process.matrix.length; i++)
{
    para (j=0; j < Process.matrix[i].length; j++)
    {
        Process.values[k] = Process.matrix[i][j]
        k++
    }
}
```

## Métodos de acesso em um campo indexado

Campos indexados suportam os seguintes métodos de matriz do JavaScript:

### **concat()**

Concatena elementos em uma matriz.

### **join()**

Converte todos os elementos da matriz para sequências de caracteres e os concatena.

### **pop()**

Remove um item do final de uma matriz.

**Observação:** se a matriz pertencer ao conjunto de dados do operador de outro operador, não use o método `pop` para matrizes do JavaScript para avaliar parâmetros do operador.

### **push()**

Envia um item para o final de uma matriz.

### **reverse()**

Inverte a ordem dos elementos em uma matriz.

### **shift()**

Altera um elemento do início de uma matriz.

### **slice()**

Retorna um fatia da sub-matriz de uma matriz.

### **sort()**

Classifica os elementos de uma matriz.

### **splice()**

Insere, exclui ou substitui os elementos da matriz.

### **toLocaleString()**

Converte uma matriz para uma sequência de caracteres localizada.

### **toString()**

Converte uma matriz para uma sequência de caracteres.

### **unshift()**

Insere os elementos no início de uma matriz.

Para obter informações sobre como usar esses métodos, consulte o Guia de Referência do JavaScript.

## **Especificar as variáveis do conjunto de dados nomeado**

Os campos nos objetos do conjunto de dados (conjuntos de dados nomeados) são identificados em expressões usando a notação de ponto (*.string*) ou de colchetes (*[expression]*):

```
Datasets[path_expression][variable_name_expression]
```

```
Datasets[path_expression].variable_name
```

### **path\_expression**

Representa qualquer expressão JavaScript que é avaliada como um descritor de caminho para um objeto do conjunto de dados na biblioteca atual do CA Process Automation. Um caminho para qualquer objeto na biblioteca começa com uma barra (/) para o elemento raiz, seguida pela hierarquia de pasta delimitada por barra e termina com o nome do objeto.

### **variable\_name\_expression**

Representa uma expressão que retorna o nome de um campo no objeto do conjunto de dados.

### **variable\_name**

Representa o nome real de um campo no objeto do conjunto de dados.

Uma das seguintes sintaxes é válida para fazer referência ao campo Y nas coordenadas do conjunto de dados nomeado localizado na pasta /MathValues:

```
Datasets["/MathValues/Coordinates"].Y
```

```
Datasets["/MathValues/Coordinates"]["Y"]
```

**Mais informações:**

[Caminhos relativos para conjuntos de dados](#) (na página 226)

**Especificar as variáveis do conjunto de dados do processo**

O conjunto de dados do processo contém variáveis que o desenvolvedor define ou que o CA Process Automation define automaticamente quando uma instância de processo é iniciada. A palavra-chave `Process` é usada para acessar as variáveis no conjunto de dados do processo. É possível usar uma notação de ponto ou de colchetes para especificar uma variável do processo em uma expressão:

```
Process.variable_name  
Process[expression]
```

A expressão pode especificar o nome da variável como uma sequência de caracteres literal no formato:

```
Process["variable_name"]
```

Por exemplo:

```
Process["StartDate"]  
Process.StartDate
```

O CA Process Automation define as seguintes variáveis do sistema de processo automaticamente quando ele é executado em um processo:

**CallerUser**

A ID do usuário que iniciou a instância do processo. Quando um processo pai usa um operador Iniciar processo para iniciar um processo filho, o processo pai envia o valor de `CallerUser` adiante. Quando um operador Iniciar processo em uma tarefa programada inicia outro processo, `CallerUser` é deixado em branco.

**DisplayName**

O nome do objeto de processo conforme visto na biblioteca.

**effectiveUser**

O proprietário atual do objeto de processo.

**EndDate**

A data em que essa instância do processo terminou, no formato:

```
MM/DD/AAAA
```

**EndTime**

A hora em que essa instância do processo terminou, no formato:

```
HH:MM:SS
```

**InstanceName**

O nome do objeto do processo original terminando em um identificador de objeto de tempo de execução exclusivo que identifica cada uma das instâncias de um processo. Por exemplo, 372 é anexado a process\_1, o que resulta em um InstanceName igual a process\_1\_372.

**InstanceUUID**

Somente para uso interno.

**ObjectID**

O identificador de objeto apenas para uso interno.

**ParentProcessROID**

A identificação exclusiva do objeto de tempo de execução para o processo pai que iniciou o processo filho.

**rootUUID**

Somente para uso interno.

**RuntimeROID**

Uma identificação exclusiva do objeto que o aplicativo anexa ao processo DisplayName para identificar cada instância do processo. Por exemplo, RuntimeROID 372 é anexado ao process\_1, o que resulta em um InstanceName igual a process\_1\_372.

**ScheduledStartTime**

A data e a hora em que o processo foi programado para iniciar.

**ServerName**

O nome do servidor associado ao touchpoint.

**ServerID**

Somente para uso interno.

**StartDate**

A data em que essa instância do processo foi criada, no formato:

MM/DD/AAAA

**StartTime**

A hora em que essa instância do processo foi criada, no formato:

HH:MM:SS

**TouchpointName**

O nome do orquestrador que está gerenciando o processo em execução.

## Especificar as variáveis do conjunto de dados do operador

Os conjuntos de dados do operador contêm variáveis que o desenvolvedor define no tempo de design ou que o CA Process Automation define no tempo de execução para um operador específico. As variáveis de tempo de design estão disponíveis imediatamente depois que o processo é iniciado. As variáveis de tempo de execução são adicionadas quando um operador é executado.

O nome do operador no conjunto de dados local no tempo de execução dos conjuntos de dados do operador, portanto, você pode usar a notação de ponto (.) ou de colchetes (*[expression]*) para especificar uma variável do operador em uma expressão:

```
Operator.field_name
```

```
Process["Operator_name"].field_name
```

```
Process["Operator_name"]["field_name"]
```

O código de execução anterior e posterior para um operador pode usar a palavra-chave OpName para acessar o nome do operador atual. É possível usar essa palavra-chave para acessar ou criar um conjunto de dados do operador. Para especificar uma variável de conjunto de dados do operador no código de execução anterior ou posterior desse mesmo operador, use a seguinte sintaxe:

```
Process[OpName].field_name
```

A tabela a seguir lista as variáveis comuns do sistema do conjunto de dados do operador que o CA Process Automation define automaticamente. Outras variáveis podem ser definidas para operadores específicos.

Variável do conjunto de dados do operador	Escopo do tempo de execução	Descrição
AgentName	durante, após	Nome da máquina associada ao touchpoint que executa o operador.
AgentID	durante, após	Somente para uso interno.
EndDate	após	A data em que o operador interrompeu a execução, no formato: MM/DD/AAAA
EndTime	após	A hora em que o operador interrompeu a execução, no formato: HH:MM:SS
Motivo	após	Uma sequência de caracteres que descreve o resultado.

Variável do conjunto de dados do operador	Escopo do tempo de execução	Descrição
ResponseCode	após	Uma sequência de caracteres que descreve o resultado.
StartDate	durante, após	A data em que o operador iniciou a execução, no formato:  MM/DD/AAAA
StartTime	durante, após	A hora em que o operador iniciou a execução, no formato:  HH:MM:SS
ServiceType	durante, após	O módulo do CA Process Automation que executou o operador.
TargetName	durante, após	O nome do destino (por exemplo, orquestrador).
TouchpointName	durante, após	O nome do touchpoint que executa o operador.
UUID	durante, após	Somente para uso interno.

## Especificar as variáveis do conjunto de dados do sistema

O conjunto de dados do sistema para um processo contém variáveis que retornam informações do sistema. A palavra-chave System é usada para acessar o conjunto de dados do sistema. É possível usar a notação de ponto (.) ou de colchetes ([*expression*]) para especificar uma variável do sistema em uma expressão:

```
System.variable_name
```

```
System[expression]
```

A seguinte sintaxe usa a notação de colchetes para especificar um nome de variável do sistema com uma sequência de caracteres literal:

```
System["variable_name"]
```

O conjunto de dados do sistema contém as seguintes variáveis:

**DATA**

A data atual, no formato MM/DD/AA.

**DAY**

Especifica o dia do mês.

**FIRSTDAYMONTH**

A data do primeiro dia do mês atual, no formato MM/DD/AA.

**FIRSTDAYNEXTMONTH**

A data do primeiro dia do mês seguinte, no formato MM/DD/AA.

**FIRSTDAYPREVMONTH**

A data do primeiro dia do mês anterior, no formato MM/DD/AA.

**HOST**

O nome do host atual.

**LASTDAYMONTH**

A data do último dia do mês atual, no formato MM/DD/AA.

**LASTDAYNEXTMONTH**

A data do último dia do mês seguinte, no formato MM/DD/AA.

**LASTDAYPREVMONTH**

A data do último dia do mês anterior, no formato MM/DD/AA.

**MONTH**

O mês atual, representado como um número com base em 0 (por exemplo, essa variável retorna 0 para janeiro).

**TIME**

A hora do dia atual, em minutos (por exemplo, essa variável retorna 600 para 10 AM).

**TIMES**

A hora atual do dia, no formato HHMM.

**TOMORROW**

A data do dia seguinte à data atual, no formato MM/DD/AA.

**WEEK**

A semana do mês.

**YEAR**

O ano atual.

## YESTERDAY

A data do dia que antecede o dia atual, no formato MM/DD/AA.

## Instruções

As expressões são frases JavaScript que são avaliadas para gerar um valor. As declarações do JavaScript executam comandos ou combinam uma ou mais expressões para fazer as coisas ou gerar valores. Um programa JavaScript é uma coleção de declarações.

Essa seção descreve brevemente a declaração de variável e a atribuição de variável, iterações e loops que são normalmente usados nos cálculos do CA Process Automation. A tabela a seguir lista as declarações JavaScript, algumas das quais não estão documentadas nesta seção.

### Instruções JavaScript

Instrução	Sintaxe	Descrição
Break	break; break label_name:	Sair de um switch ou declaração iterativa; ou sair da declaração nomeada por uma declaração de rótulo.
caso	expressão de caso:	Rótulos de uma declaração dentro de uma declaração switch.
continuar	continue; continue label_name:	Reinicie o loop ou o loop nomeado por uma declaração de rótulo.
padrão	padrão;	Rótulo da declaração padrão dentro de uma declaração switch.
fazer/enquanto	fazer declaração enquanto (expressão)	Executar expressões em uma declaração enquanto até uma expressão ser avaliada como falsa.
vazio	;	Não executar nenhuma ação.
para	para declaração (inicializar; testar; incremento)	Loop enquanto um teste é verdadeiro.
para/em	para declaração (variable_in_object) Loop entre as propriedades de um objeto.	Consulte "A declaração de loop para/em."

Instrução	Sintaxe	Descrição
função	function function_name(a1,a2,...an) {statements}	Declara uma função. Consulte “Incluir recursos comuns em scripts do CA Process Automation”.
if/else	if (expressão) statement1 else statement2	Executar condicionalmente. Consulte “A declaração if”.
rótulo	identificador: declaração	Atribuir um identificador para uma declaração.
return	return[ expression];	Retorno de uma função ou retornar um valor de uma função.
alternar	switch (expression) { statements }	Ramificação condicional de várias vias para declarações de caso ou padrão. Consulte “A declaração alternar”.
acionar	expressão acionar;	Aciona uma exceção.
try	try { statements }	Capturou uma exceção.
var	var name_1[=value1][, ..., name_n [=value_n]];	Declarar e, opcionalmente, inicializar variáveis. Consulte “Declaração de variável”.
while	while (expression) statement	Executar expressões em uma declaração enquanto até uma expressão ser avaliada como verdadeira. Consulte “A declaração de loop enquanto”.

**Mais informações:**

[Declaração de variável](#) (na página 266)

[A declaração if](#) (na página 267)

[A declaração switch](#) (na página 269)

[A declaração while loop](#) (na página 270)

[A declaração for/in Loop](#) (na página 271)

[Incluir recursos comuns em scripts do CA Process Automation](#) (na página 274)

## Declaração de variável

A instrução `var` do JavaScript declara uma variável de JavaScript. Opcionalmente, você pode usar o operador de atribuição `=` para inicializar uma variável ao mesmo tempo em que a cria. A definição de variável de JavaScript usa a seguinte sintaxe:

```
var variable_name [= initial_value];
```

As seguintes linhas criam variáveis, mas deixam os valores iniciais indefinidos até que o código subsequente atribua valores a elas:

```
var x  
var s
```

É possível inicializar uma variável como um número inteiro ou uma sequência de caracteres. No exemplo a seguir, `x` é inicializado como um número inteiro e `s` é inicializado como uma sequência de caracteres vazia.

```
var x = 0  
var s = ""
```

É possível criar diversas variáveis em uma única instrução:

```
var i = 0, j = 0, k = 0
```

## Atribuição de variável

Para atribuir valores às variáveis do conjunto de dados, usar o operador de atribuição. A atribuição de variável usa a seguinte sintaxe:

```
[dataset_reference.]variable_name = expression;
```

A expressão consiste em qualquer combinação de funções, variáveis, valores e operadores que retornam uma sequência de caracteres ou um valor de número inteiro. Por exemplo:

```
Process.S = "ABCDEF" + '_' + "123"
```

```
Datasets["ThisDataset"].x = 18 * I
```

```
x = 18 * I
```

Se você omitir a referência do conjunto de dados, o JavaScript fará referência ao conjunto de dados do processo, por padrão. Se a variável não existe no conjunto de dados do processo, o JavaScript criará uma variável temporária. Para criar uma variável do conjunto de dados do processo, use a referência do processo.

## Reutilizar variáveis

Seleção da variável do processo pai-filho: capacidade que um processo filho tem de consultar as variáveis inicializadas pelo processo pai

## Declarações condicionais

As expressões do CA Process Automation reconhecem as declarações condicionais do JavaScript. A seleção da condicional if avalia uma única condição booliana enquanto a condição else if avalia uma série de condições booleanas. As declarações condicionais do CA Process Automation esperando um valor booliano reconhecer 0 como falso e qualquer número inteiro não zero como verdadeiro. As expressões do CA Process Automation também oferecem suporte a declaração switch, o que permite que vários resultados ao avaliar uma única variável.

### A declaração if

A declaração de seleção condicional if usa a seguinte sintaxe:

```
if (Boolean_expression)
```

```
    Instrução
```

O Boolean\_expression é qualquer combinação de funções, variáveis, valores e operadores que retornam um único valor verdadeiro ou falso. Por exemplo:

```
if (i <= 18) {  
    y = 18 * I  
    z = y * 56  
}
```

O segundo formulário da declaração de seleção condicional if permite dois resultados da expressão booliana. Usa a seguinte sintaxe:

```
if (Boolean_expression)  
    statement1  
else  
    statement2
```

Por exemplo:

```
if (i <= 18)  
    Process.Date = System.Date  
else  
    Process.date ="2006/01/23"
```

## A declaração else if

Para vários resultados, você pode aninhar declarações if/else. No entanto, a lógica pode se tornar complexa para seguir com muitos aninhamentos. Portanto, é possível usar a seguinte construção para uma série de declarações if/else:

```
if (Boolean_expression_1)
    statement_1
else if (Boolean_expression_2)
    statement_2
else if (Boolean_expression_3)
    statement_3
...
else if (Boolean_expression_n)
    statement_n
else
    statement_else
```

A declaração else final é opcional. Ele apenas especifica o código a ser executado se nenhuma das expressões booleanas forem verdadeiras.

## A declaração switch

A declaração switch executa um ramificação de várias vias, úteis quando todas as ramificações de uma declaração condicional dependem da mesma variável. Nesse caso, é complexo para verificar o valor da mesma variável repetidamente usando várias declarações if. A declaração switch usa a sintaxe a seguir para fazer a mesma coisa de forma mais eficiente:

```
switch(variável)
{
    case value_1:
        Instruções
    break;
    case value_2:
        Instruções
    Break
    ...
    case value_n:
        Instruções
    Break
    padrão:
        Instruções
    Break
}
```

A declaração switch executa o código na declaração de caso que corresponde ao valor atual da variável. Se não houver correspondência, a declaração switch executa o código padrão ou passa para a próxima declaração se não houver um código padrão. As declarações break delimitam opcionalmente um bloco de código do caso no próximo caso. Na ausência de uma declaração break, a execução vai de um caso para o próximo. Essa é uma ação legal, tenha cuidado para não omitir uma declaração break a menos que você realmente planejada executar para passar para a próxima declaração de caso.

## Declarações iterativas

O JavaScript tem várias declarações de loop iterativas, uma declaração contínua e uma declaração break. As while e do-while loops executam uma ou mais declarações desde que alguma condição seja verdadeira. As for e for loops executam uma ou mais declarações um número de vezes especificado. A declaração break sai de uma declaração iterativa. A declaração continue reinicia um loop em uma nova iteração.

## A declaração while loop

A while loop tem a seguinte sintaxe:

```
while (Boolean_expression)
    Instrução
```

O while loop executa uma sequência de instruções desde que a expressão booliana testada no início do loop retorne um valor verdadeiro. Por exemplo:

```
var n = 0
while (n < 10)
{
    Process.square[n] = n * n
    n++
}
```

## A declaração do/while loop

O do-while loop tem a seguinte sintaxe:

```
fazer
    Instrução
while (Boolean_expression);
```

O do-while loop é semelhante ao while loop, exceto pelo fato de que ele testa na parte inferior do loop, em vez de no início do loop. O while loop executa uma sequência de declarações desde que a expressão booliana retorne um valor verdadeiro. Por exemplo:

```
var n = 0
do {
    Process.square[n] = n * n
} while (n++ < 10)
```

## A declaração loop

O for loop executa uma sequência de declarações por um número de vezes especificado. A for loop tem a seguinte sintaxe:

```
for (inicializar; testar; incrementar)
    Instrução
```

O for loop é semelhante ao while loop, exceto pelo fato de que um incremento e uma inicialização estão incluídos na sintaxe de loop. Cada iteração do for loop aumenta o incremento, executa o teste e executa a declaração.

Por exemplo, em uma variável indexada Process.square contendo 35 valores, é possível usar as seguintes linhas de código para definir cada valor para o quadrado de seu índice:

```
for (var i = 0; i < 34; i++)
    Process.square[i] = i * i
```

## A declaração for/in Loop

A instrução for/in loop executa uma sequência de *instruções* para todos os valores de uma *variável* especificada em um *objeto* especificado. A for/in loop tem a seguinte sintaxe:

```
for (variável no objeto)  
    instrução
```

O valor da *variável* é um dos seguintes itens:

- O nome de uma variável
- Uma instrução var que declara uma variável
- Um elemento de matriz
- Uma propriedade do objeto.

Em outras palavras, o valor da *variável* é equivalente ao lado esquerdo de uma expressão de atribuição. O valor do *objeto* é o nome de um objeto ou uma expressão que é avaliada para um objeto.

Por exemplo, é possível fazer o loop por meio de elementos de um campo indexado em um conjunto de dados, defina uma variável de índice e especifique o campo indexado como o objeto.

```
para (var i no Process.square)  
    Process.square[i] = i * i
```

## A declaração break

A declaração break pode ser usada para sair de um loop, como ilustrado nas seguintes linhas de código.

```
var l = 0;  
while (l < 10) {  
    n = n++;  
    if (n > 102)  
        break;  
}
```

## A declaração continue

A declaração continue pode ser usada para passar para a próxima iteração de um loop. O seguinte (em vez de trivial) exemplo ilustra o uso da declaração contínua para atribuir números pares para uma variável local (Processo) indexada.

```
var i = 0, j = 0
for (j=0; j < 102; j++)
{
    if (j%2) continue
    // seguindo a declaração executada apenas para valores pares de j
    Process.evans[i] = j
    i++
    // seguindo interrompe o loop quando todos os elementos da matriz são
    concluídos
    if (i >= Process.evans.Size) break
}
```

## Especificar caminhos do sistema em expressões do CA Process Automation

Os cálculos geralmente aceitam caminhos UNIX ou Microsoft Windows. O caminho UNIX funciona para locais nos sistemas host UNIX e Microsoft Windows. Por exemplo:

```
Process.Path = "/tmp/files/myfile"
```

O exemplo anterior especifica o local na unidade atual para um host Microsoft Windows ou a raiz de um host UNIX. Especifique um caminho de rede, como segue:

```
Process.NetPath = "//myhost/tmp/files/myfile.txt"
```

Incluir a especificação de unidade em um caminho para um sistema Microsoft Windows como segue:

```
Process.Path = "C:/tmp/files/myfile.txt"
```

Se um diretório de trabalho (como C:\tmp) é especificado para um processo do Microsoft Windows, especifique um caminho no diretório de trabalho sem qualquer barra no início, como segue:

```
Process.Subdir = "files/myfile.txt"
```

Se usar um caminho do Microsoft Windows em um cálculo, verifique de tirar o caractere barra invertida, de modo que o intérprete avalie corretamente como um caractere literal, como segue:

```
Process.Path = "C:\\tmp\\files\\myfile.fm"
```

## Incluir recursos comuns em scripts do CA Process Automation

É possível incluir scripts definidos anteriormente em um script do CA Process Automation. Isso permite que um script leia e acesse as funções salvas em tempo de execução. Use a declaração `include` em qualquer linha de uma caixa de diálogo do script para adicionar um script salvo anteriormente para o arquivo. A declaração `include` usa a seguinte sintaxe:

```
include(expression)
```

O argumento da expressão pode ser qualquer caminho que faz referência a um recurso apropriado. Os caminhos reconhecidos incluem:

- Um caminho relativo, como incluir ("`Scripts/functions.js`"), especifica um recurso de usuário comum (`c2ouserresources`) no repositório do CA Process Automation.

**Observação:** para obter mais informações sobre como adicionar ou gerenciar recursos no repositório do CA Process Automation, consulte "*Gerenciar recursos comuns*" no *Guia de Administração*.

- Um caminho de diretório, como incluir ("`/scripts/functions.js`") especifica um script na unidade atual para um host do Microsoft Windows ou a raiz de um host UNIX.
- Incluindo a letra da unidade em um caminho explícito, como incluir ("`D:\scripts\functions.js`"), especifica um script em uma unidade especificada.
- Um caminho de rede, como incluir ("`//share/scripts/function.js`"), especifica um script em um recurso de rede compartilhado.
- Um URL, como incluir ("`http://james:8080/itpam/scripts/functions.js`"), especifica um caminho para um recurso da web.

Linhas em um script incluído são adicionadas a um script como se eles fossem digitados no lugar da declaração `include`. Observe que ele é uma prática recomendada para incluir somente funções necessárias ou outro código em vez de grandes bibliotecas de funções no script incluído. Scripts incluídos são compilados no tempo de execução, muitas linhas de código não usadas aumenta desnecessariamente o tempo necessário para executar um script.

## Comentários nos cálculos do CA Process Automation

Os comentários de JavaScript são separados em linhas pelo caractere `//`. O início de um comentário sempre significa o final de uma linha lógica. Um comentário é iniciado no final da linha lógica e termina no final da linha física. Os comentários são ignorados pelo interpretador de idioma do JavaScript.

# Capítulo 7: Formulários

---

O CA Process Automation oferece suporte a dois tipos principais de objetos de formulário interativos:

- Formulários de solicitação inicial
- Formulários de solicitação de interação

Crie esses formulários em pontos estratégicos de seu processo para permitir que os usuários forneçam entradas e controlem o processo.

Operadores personalizados também incluem formulários com páginas e campos de dados que são exibidos na paleta Propriedades do Criador de processos.

Esta seção contém os seguintes tópicos:

[Formulários de solicitação inicial](#) (na página 275)

[Formulários de solicitação de interação](#) (na página 277)

[Criador de formulários](#) (na página 279)

[Inicializar variáveis do formulário](#) (na página 367)

## Formulários de solicitação inicial

O objeto Formulário de solicitação inicial permite que você crie uma interface que, por sua vez, permite que outros usuários iniciem um processo e forneçam uma entrada na inicialização de uma maneira estruturada. Você cria e mantém o layout e o comportamento do formulário. Os usuários preenchem o formulário quando solicitado. Por exemplo, é possível fornecer a uma pessoa do suporte técnico ou de recursos humanos a capacidade de fornecer informações que influenciem como um processo relacionado é iniciado.

Você pode agrupar uma série de elementos do formulário relacionados em uma página específica ou em uma seção específica da página. É possível adicionar qualquer número de páginas. Você pode usar as funções e os eventos para obter ou definir outros valores de campo. Crie o formulário para coletar todas as informações necessárias do usuário para iniciar o processo.

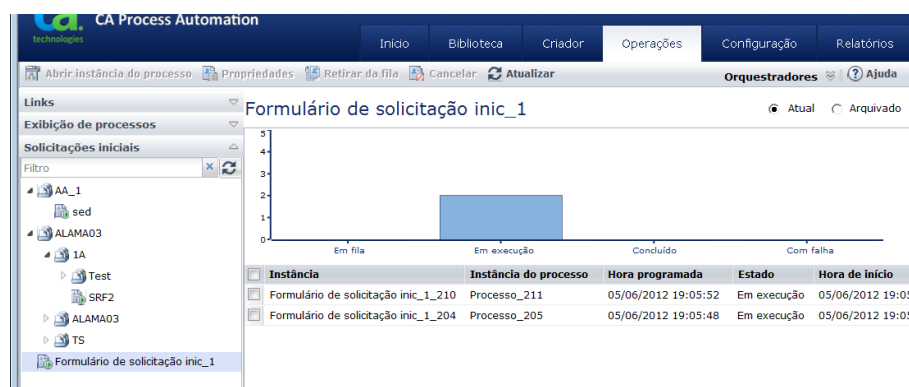
## Monitorar instâncias de formulário de solicitação inicial e instâncias de processo

Após criar um formulário de solicitação inicial, ele deverá ser iniciado como parte de outro processo, ou poderá ser iniciado manualmente. Quando um formulário é iniciado, resulta em uma nova instância de formulário com um nome exclusivo que consiste no nome do formulário mais a ID do objeto de tempo de execução do formulário. Quando um processo é iniciado, resulta em uma nova instância de processo com um nome exclusivo que consiste no nome do processo mais a ID do objeto de tempo de execução do processo. Os exemplos estão a seguir:

```
MyStartRequestForm_239  
MyProcess_241
```

### Siga estas etapas:

1. Clique na guia Operações.
2. No painel Operações, vá para qualquer um dos seguintes locais para exibir as instâncias de formulário e de processo:
  - a. Expanda Solicitações iniciais, expanda as pastas e, em seguida, clique em um formulário de solicitação inicial.
  - b. Expanda Exibição de processos, expanda as pastas, expanda um objeto Exibição de processos e, em seguida, clique em um formulário de solicitação inicial opcional que tenha sido adicionado ao objeto anteriormente.
  - c. Expanda Links e clique em Solicitações iniciais.
3. No painel Operações, vá até um dos seguintes locais para iniciar um formulário e seu processo associado:
  - a. Expanda Solicitações iniciais, expanda as pastas e, em seguida, clique em um formulário de solicitação inicial. Clique com o botão direito do mouse no formulário e selecione Iniciar.
  - b. Expanda Exibição de processos, expanda as pastas, expanda um objeto Exibição de processos e, em seguida, clique em um formulário de solicitação inicial opcional que tenha sido adicionado ao objeto anteriormente. Clique com o botão direito do mouse no formulário e selecione Iniciar.
4. Repita a etapa 2 para exibir a instância do formulário e a instância do processo.
5. Para monitorar a instância de processo real, clique em uma linha e, em seguida, clique em Abrir instância do processo na barra de ferramentas.



### Observações:

- A coluna Instância do processo não mostra nenhum dado de formulários no estado *Em fila*. Formulários em fila não criam uma instância de processo até que o formulário esteja *Em execução*.
- É possível alterar o nome da instância do processo enquanto o processo estiver em execução usando `Process.UserName`. Clique em *Atualizar* para exibir o novo nome da instância do processo.
- A coluna Instância do processo não inclui os formulários que já estavam em execução antes da atualização para a versão atual do CA Process Automation.

## Formulários de solicitação de interação

O Formulário de solicitação de interação permite criar uma interface que pode ser usada durante a execução de um processo para interagir com um usuário de uma maneira estruturada. O formulário é acessado em um navegador web por um usuário que administra os processos dentro do CA Process Automation ou executa algum outro objetivo de negócio.

Os casos de uso típico para os formulários de solicitação de interação incluem:

- Obter aprovação antes de continuar com um processo ou um caminho em um processo.
- Permitir que um usuário selecione um curso de ação.
- Recuperar informações que estão disponíveis somente para uma pessoa em tempo de execução.
- Solicitar ações manuais (por exemplo, conectar fisicamente um servidor a um switch) a serem executadas e marcadas como concluídas antes de continuar com um processo.

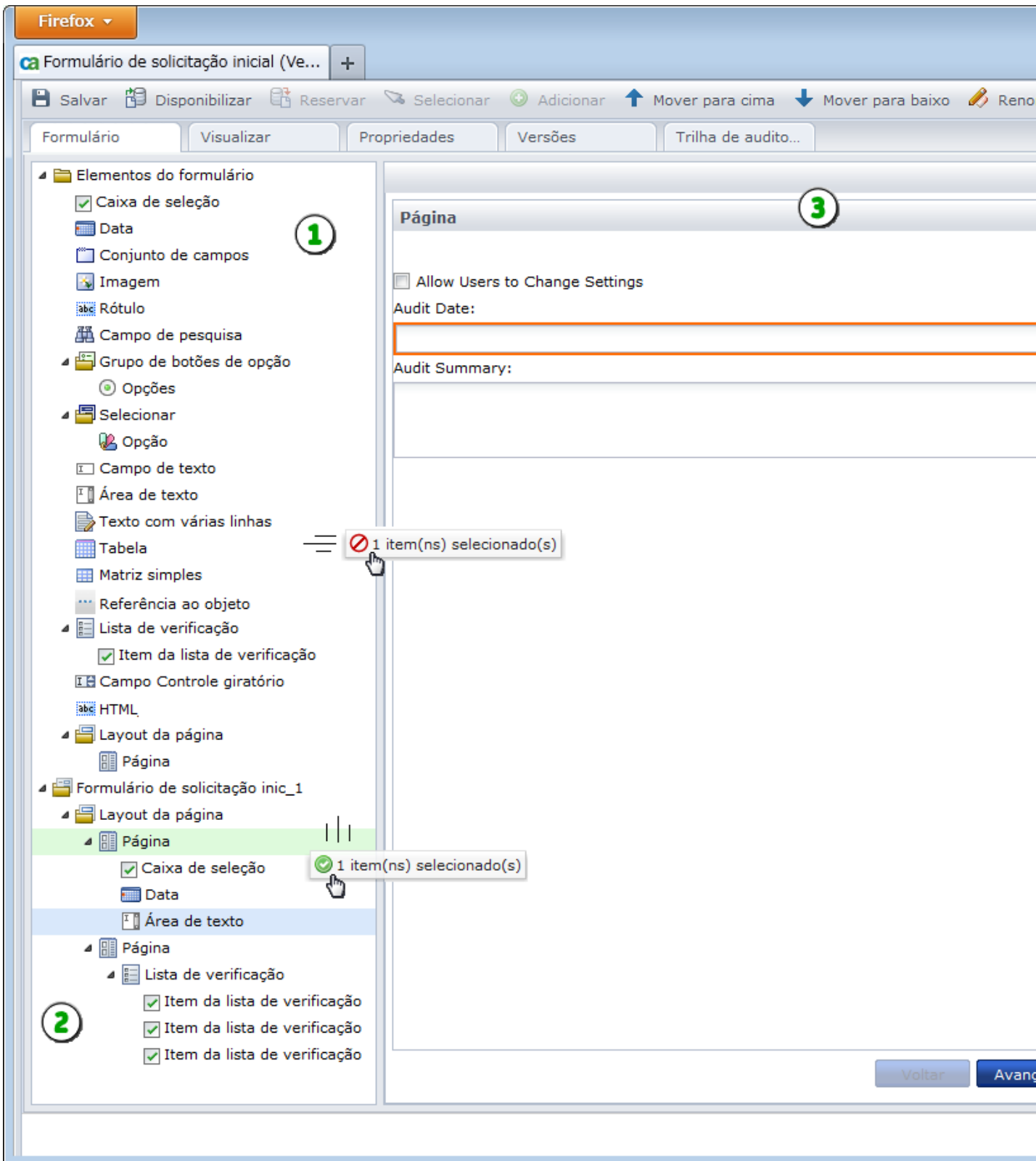
O objeto Formulário de solicitação de interação define as páginas, os parâmetros e outras características do formulário. Os parâmetros podem ser configurados para exibir caixas de edição, listas suspensas, caixas de lista e listas de caixas de seleção em páginas de formulários. É possível ter várias páginas no formulário, permitindo definir o grupo de parâmetros relacionados em páginas separadas. Os usuários clicam nos botões voltar e avançar para percorrer entre as páginas em um formulário de solicitação de interação. É possível adicionar qualquer número de páginas a um Formulário de solicitação de interação para reunir todas as informações necessárias durante a execução de um processo que usa o formulário.

Os Formulários de solicitação de interação são salvos como objetos separados na biblioteca de automação. Após a criação e a disponibilização de um Formulário de solicitação de interação, ele poderá ser adicionado a qualquer processo usando o operador Atribuir tarefa de usuário.

Quando um processo executa o operador Atribuir tarefa de usuário, o Formulário de solicitação de interação é listado como um prompt de usuário pendente no CA Process Automation. O operador Interação do usuário não é concluído até que um usuário autorizado responda ao prompt preenchendo e enviando o Formulário de solicitação de interação ou até que o tempo limite especificado no operador Atribuir tarefa de usuário seja atingido.

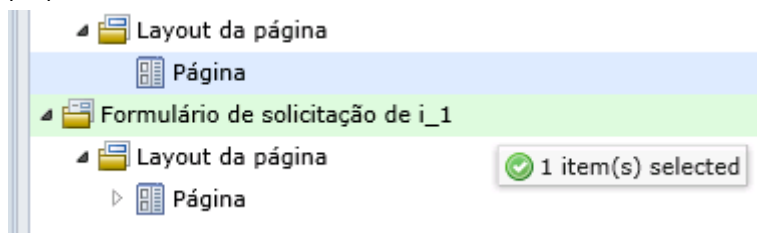
## Criador de formulários

Ao abrir um formulário no navegador da biblioteca, o Criador de formulário é exibido. O Criador de formulário inclui uma barra de ferramentas padrão e uma série de guias. Use a guia Formulário para criar um formulário e use a guia Visualizar para analisar sua aparência.



---

Item:	Descrição:
1	<b>Elementos do formulário:</b> a parte superior desse painel exibe todos os tipos disponíveis de controles.
2	<b>Estrutura do formulário:</b> a parte inferior desse painel exibe a estrutura do seu formulário. Arraste e solte os elementos do formulário nas páginas do formulário a partir daqui.
3	<b>Páginas do formulário:</b> o layout das páginas do formulário é exibido aqui. Clique em um controle para editar suas propriedades. Para formulários com várias páginas, clique em Voltar e em Avançar para exibir as outras páginas. Os usuários também podem clicar em Voltar e em Avançar para exibir as páginas do formulário.  Arraste e solte elementos do formulário na página para criar os formulários. <b>Observação:</b> a área que está disponível para soltar uma página em outra página é muito pequena. Recomendamos arrastar o elemento de página até a área lateral entre os elementos do formulário e o próprio formulário:



---

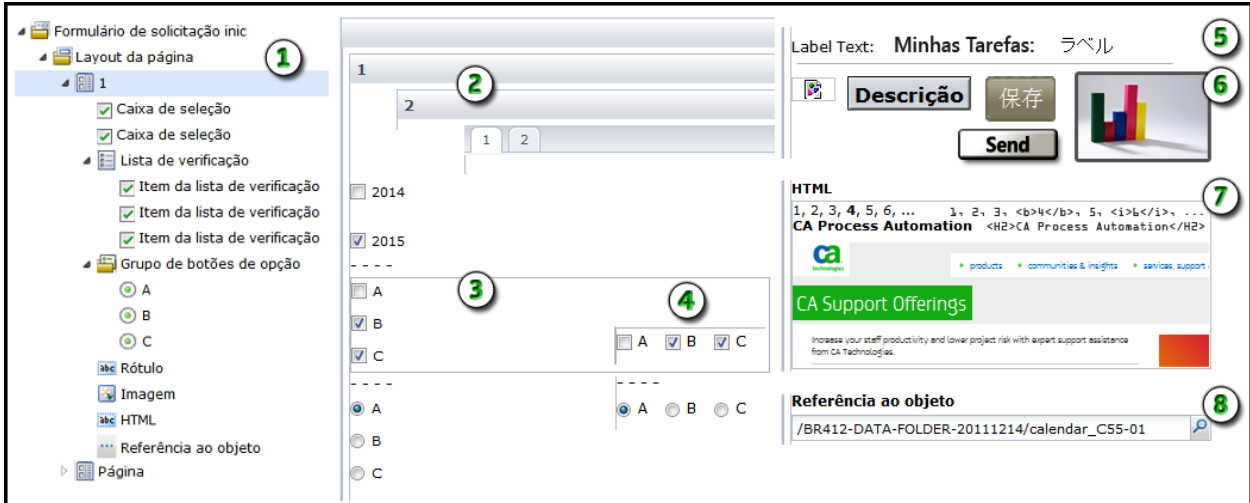
4	<b>Painel Propriedade:</b> use esse painel para exibir ou editar as variáveis nos elementos do formulário. Por exemplo, defina a propriedade Obrigatório como verdadeiro, altere o rótulo que identifica um campo ou especifique uma função para um evento.  A ilustração mostra os seguintes argumentos usados no evento onFocus para definir o valor de um campo de texto chamado Form.Var_3 como 500:
---	--

```
ca_pam_setTextFieldValue(Form.Var_3, 500)
```

---

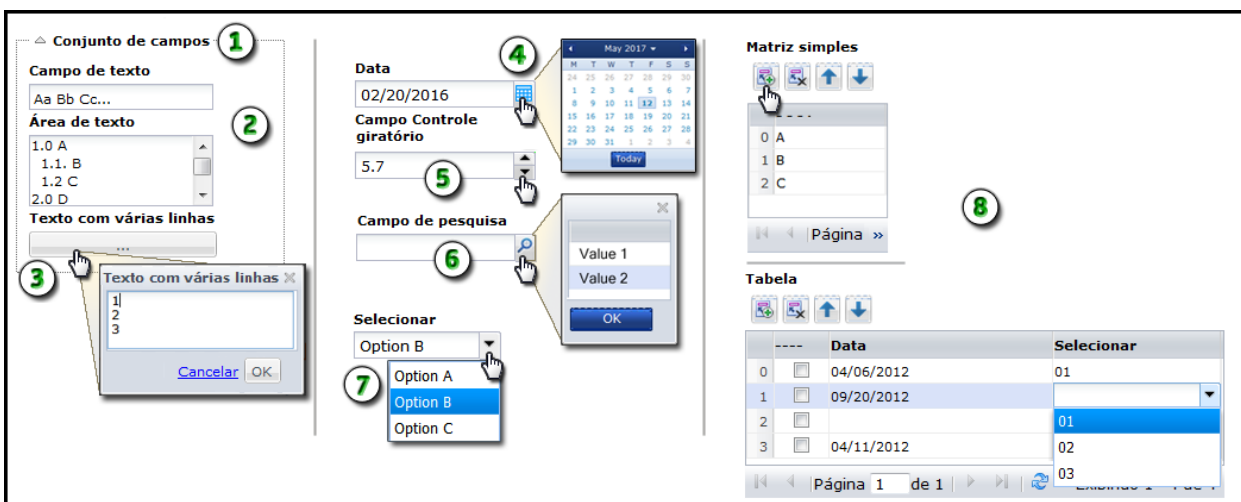
## Elementos do formulário

Este tópico apresenta exemplos básicos de cada tipo de elemento do formulário.



**Item:**      **Descrição:**

- 1**      **Estrutura do formulário:** ao criar o formulário, a parte inferior da guia Formulário exibe a organização das páginas e dos elementos do formulário.
- 2**      **Layout da página:** é possível configurar o layout para exibir páginas como cartões ou guias. Os *cartões* são exibidos em sequência quando o usuário clica em Voltar ou Avançar. As *guias* permitem que o usuário selecione qualquer guia para exibir a página associada.
- 3**      **Caixas de seleção e botões de opção:** use um grupo de *caixas de seleção* (itens da lista de verificação) quando um usuário puder selecionar vários itens relacionados. Use um grupo de *botões de opção* quando um usuário puder selecionar apenas um item relacionado. Use caixas de seleção individuais para controlar as configurações de itens não relacionados.
- 4**      **Orientação:** é possível definir a orientação da lista de verificação e do grupo de botões de opção para vertical ou horizontal.
- 5**      **Rótulos:** use *rótulos* para identificar campos ou regiões específicas do formulário.
- 6**      **Imagens:** use um elemento de *imagem* para especificar um gráfico. Uma imagem pode exibir um logotipo, um ícone, um indicador de status ou um botão no qual os usuários podem clicar. Quando o formulário não conseguir localizar uma imagem, o produto exibirá um ícone de link quebrado.
- 7**      **HTML:** use o elemento *HTML* para especificar o código HTML a ser processado para o usuário.
- 8**      **Referência ao objeto:** use uma *referência ao objeto* para oferecer aos usuários um modo fácil de selecionar outro objeto no navegador da biblioteca. Uma referência ao objeto armazena o caminho para um objeto em uma biblioteca. Por exemplo, um usuário pode especificar um touchpoint em um orquestrador e, em seguida, executar um processo no touchpoint selecionado. Como um criador de formulário, você pode limitar os tipos de objeto disponíveis que uma referência ao objeto permite.




---

**Item:**    **Descrição:**


---

- 1**    **Conjunto de campos:** use *conjuntos de campos* para agrupar elementos do formulário relacionados. Os usuários podem expandir e recolher conjuntos de campos para concentrar-se em partes específicas de um formulário, evitando confusão.

---
- 2**    **Campo de texto e Área de texto:** use os *campos de texto* para que os usuários possam inserir dados básicos, como nomes, endereços, contas de email, números de telefone e outros detalhes. Use as *áreas de texto* para que os usuários possam inserir várias linhas de texto. Os usuários podem exibir a quantidade de texto que você definir na propriedade de altura do formulário e rolar para exibir o restante do campo.

---
- 3**    **Texto de várias linhas:** use um *texto de várias linhas* para exibir grandes quantidades de texto em uma nova janela de rolagem redimensionável. Esse elemento ocupa uma única linha no formulário. O formulário exibe o elemento como um botão com reticências (...) para indicar que o usuário pode clicar para explorar a janela inteira.

---
- 4**    **Campo de data:** use um *campo de data* para armazenar uma data. Os usuários podem digitar ou selecionar uma data no controle de calendário integrado.

---
- 5**    **Controle giratório:** use um *controle giratório* para permitir que o usuário ajuste um valor para cima ou para baixo em incrementos predefinidos.

---
- 6**    **Pesquisa:** use um *elemento de pesquisa* para exibir uma tabela com uma coluna de valores em uma janela pop-up. Você pode usar um conjunto de dados ou uma origem de dados externa para fornecer os valores. Um usuário pode clicar em um valor e, em seguida, clicar em OK para preencher o campo de pesquisa e armazenar o valor.

---
- 7**    **Selecionar:** use um elemento de *seleção* para exibir uma lista suspensa de opções.

---
- 8**    **Matriz simples e Tabela:** use uma *matriz simples* para armazenar um único tipo de dado em uma tabela. Use uma *tabela* para armazenar várias colunas de dados. Esses elementos do formulário incluem opções para adicionar, excluir e mover as linhas.

---

Você pode inserir os seguintes elementos em uma tabela:

- Caixa de seleção
- Data
- Conjunto de campos
- Imagem
- Rótulo
- Campo de consulta
- Botão de opção
- Campo de seleção
- Campo de texto
- Área de texto
- Texto de várias linhas
- Table
- Matriz simples
- Referência ao objeto
- Lista de verificação
- Campo giratório
- HTML
- Página

## Propriedades do elemento do formulário

**Observação:** aponte para um atributo de elemento do formulário para exibir a descrição da dica de ferramenta do elemento a fim de obter ajuda no preenchimento do formulário.

### Permitir adição de linhas

Um valor booleano (verdadeiro ou falso) para tabelas e matrizes simples. Quando verdadeiro, um botão permite que os usuários do formulário criem linhas na tabela. Quando falso, os usuários não podem adicionar linhas à tabela.

### Permitir números decimais

Um valor booleano (verdadeiro ou falso) para campos do controle giratório. Quando verdadeiro, os usuários podem inserir números com decimais, como 12,25 ou 0,003. Quando falso, os usuários não podem digitar números decimais no campo; apenas números inteiros.

**Permitir exclusão de linhas**

Um valor booleano (verdadeiro ou falso) para tabelas e matrizes simples. Quando verdadeiro, um botão permite que os usuários do formulário removam linhas da tabela. Quando falso, os usuários não podem remover as linhas da tabela.

**Permitir números negativos**

Um valor booleano (verdadeiro ou falso) para campos do controle giratório. Quando verdadeiro, os usuários podem inserir números menores que zero, como -10. Quando falso, os usuários não podem digitar números negativos; apenas 0 ou números positivos.

**Permitir reordenar linhas**

Um valor booleano (verdadeiro ou falso) para tabelas e matrizes simples. Quando verdadeiro, dois botões permitem que os usuários do formulário movam as entradas para cima e para baixo na tabela. Quando falso, os usuários não podem mover as linhas para cima ou para baixo na tabela.

**Calendário**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Calendário. Quando falso ou em branco, o tipo Calendário do objeto de automação não é exibido. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

**Rótulo da caixa de seleção**

A sequência de caracteres de texto ou o nome aplicado a uma única caixa de seleção ou a um item de caixa de seleção em um grupo.

**Criado por**

O nome do usuário ou a conta de usuário que criou o formulário.

**Criado em**

A data e a hora em que o formulário foi criado.

### **Ícone personalizado**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Ícone personalizado. Quando falso ou em branco, o tipo Ícone personalizado do objeto de automação não é exibido. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### **Operador personalizado**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Operador personalizado. Quando falso ou em branco, o tipo Operador personalizado do objeto de automação não é exibido. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### **Conjunto de dados**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Conjunto de dados. Quando falso ou em branco, o tipo Conjunto de dados do objeto de automação não é exibido. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### **Expressão de conjunto de dados**

Uma referência a uma variável de matriz específica em um objeto Conjunto de dados na biblioteca, conforme o exemplo a seguir:

```
Conjuntos de dados ["/MyFolder/MyForm/MyDatasetObject"].MyArray
```

### **Formato da data**

O formato preferencial para datas. Por exemplo, insira o valor dd de MM de aaaa para formatar a data 01/05/2014 como 01 de maio de 2014. Digite aaaaMMdd para exibir a mesma data como 20140501. O formato de data padrão é dd/MM/aaaa.

Você também pode armazenar a hora em um campo Data. Por exemplo, você pode definir a propriedade Formato de data como dd/MM/aaaa hh:mm:ss para exibir o valor de data e hora como 01/05/2014 10h17min43s.

### **Descrição**

Informações sobre o elemento do formulário, além de seu nome ou rótulo resumido.

**Desabilitado**

Uma configuração opcional que pode ser aplicada a um elemento de formulário. Os seguintes comportamentos caracterizam os elementos com a propriedade Desativado definida:

- O elemento do formulário não pode receber o foco.
- O elemento do formulário é exibido esmaecido.
- Não é possível alterar o valor de um elemento de formulário desativado. No entanto, você pode copiar o valor para reutilizá-lo.

Desative um elemento do formulário para aplicar a lógica de negócios e evitar dados inválidos.

**Editável**

Um valor booleano (verdadeiro ou falso) para um campo. Quando verdadeiro, os usuários podem editar o campo. Quando falso, os dados são somente leitura.

**Altura**

A quantidade de espaço vertical ocupado pelo elemento do formulário. Especifique um valor (em pixels) para essa propriedade, a partir da parte superior até a parte inferior do elemento do formulário.

**Oculto**

Um valor booleano (verdadeiro ou falso) para um campo.

**Verdadeiro**

Torna o campo invisível quando o formulário for processado.

### Ocultar o rótulo

Um valor booleano (verdadeiro ou falso) para um campo ou uma tabela com um rótulo. Quando verdadeiro, o rótulo é ocultado. Quando falso ou em branco, o rótulo fica visível.

### \_id

Um identificador exclusivo somente leitura para uma instância específica de um elemento do formulário. O \_id anexa os seguintes itens ao nome do formulário com um ponto (.) como separador:

- O nome de qualquer objeto pai, como uma tabela.
- O nome do elemento do formulário.

Por exemplo, Form1 tem dois grupos de botões de opção com as seguintes propriedades de \_id:

```
Form1.rgName1  
Form1.rgName2
```

Um botão de opção específico pode ter o seguinte \_id:

```
Form1.rgName2.RadioOptionA
```

Um elemento do formulário em uma tabela chamada Table\_2 pode ter o seguinte \_id:

```
Form_1.Table_2.Var_3
```

O \_id de um formulário é igual ao valor de sua propriedade Nome.

**Observação:** o \_id é utilizado em funções JavaScript. Ao renomear ou mover um elemento, esteja ciente de que você também está alterando seu \_id.

### Aumentar

Para campos do controle giratório, o valor de um ajuste incremental, para cima ou para baixo. Por exemplo, se o campo exibir 6,55 e o incremento for de 0,02, um clique para cima no valor resultaria em 6,57 e um clique para baixo resultaria em 6,53.

### Formulário de solicitação de interação

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Formulário de solicitação de interação. Quando falso ou em branco, os objetos de automação do tipo Formulário de solicitação de interação não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

**Invisível**

Torna um campo visível ou invisível mas somente leitura apenas nos formulários do operador personalizado.

**Verdadeiro**

Torna o campo invisível apenas em operadores personalizados.

**Observação:** esse atributo é válido somente em operadores personalizados. O atributo Oculto apenas oculta um campo, enquanto o atributo Invisível oculta o campo e também define o campo como somente leitura.

Nos operadores personalizados, o atributo Invisível criado no CA Process Automation release 3.x e o atributo Oculto criado no CA Process Automation release 4.1 estão mapeados para esse atributo no CA Process Automation release 4.2.

**Rótulo**

O nome descritivo curto de um elemento de formulário exibido ao usuário.

**Alinhamento de rótulo**

A propriedade somente leitura que indica como os rótulos de campo são exibidos em relação aos elementos do formulário que eles descrevem. No Criador de formulário, você pode clicar nos botões da barra de ferramentas para alinhar os rótulos à esquerda, à direita e acima dos elementos do formulário selecionado.

**Largura do rótulo**

O tamanho (em pixels) de uma linha de texto do rótulo. Os rótulos longos são quebrados para a próxima linha.

**Layout**

Uma propriedade apenas de criação que determina a maneira como várias páginas em um formulário são exibidas nas guias Formulário e Visualizar.

**Cartão**

Exibe uma página por vez com os botões Voltar e Avançar para navegar pelas páginas em uma sequência. É desse modo que as páginas de todos os formulários são exibidas aos usuários do formulário. Cartão é a configuração de Layout padrão.

**Guia**

Exibe os nomes das páginas disponíveis em suas próprias guias. Por motivos de conveniência, os criadores podem clicar em uma guia para ir para a página associada em qualquer ordem.

**Tamanho máximo**

O número máximo de caracteres que um usuário pode digitar no campo. Por exemplo, para exigir um número de 8 dígitos em um campo específico, defina as propriedades Tamanho máximo e Tamanho mínimo do campo como 8.

**Número máximo de linhas**

O maior número de entradas permitidas em uma tabela.

**Tamanho mínimo**

O número mínimo de caracteres que um usuário pode digitar no campo. Por exemplo, para exigir um número de 8 dígitos em um campo específico, defina as propriedades Tamanho máximo e Tamanho mínimo do campo como 8.

**Mínimo de linhas**

O menor número de entradas permitidas em uma tabela.

**Modificado por**

O nome do usuário ou a conta de usuário que modificou o formulário.

**Modificado em**

A data em que o objeto do formulário foi alterado pela última vez.

**Nome**

Uma sequência de caracteres exclusiva que identifica um elemento do formulário. O sistema atribui um nome inicial, como Var\_3, que você pode modificar. A alteração do valor do nome também altera o valor de `_id`.

Os elementos do formulário podem ter dois nomes separados: uma propriedade Nome e um nome interno que identifica o elemento do formulário no Criador de formulários.

- Altere o valor de Nome para definir o valor da variável `_id` que identifica o elemento do formulário.
- Clique em Renomear para mudar o nome interno de um elemento que é exibido no layout da estrutura hierárquica do formulário no momento da criação. Esse nome interno é o valor padrão fornecido ao Rótulo de um elemento do formulário. Esse rótulo é exibido para os usuários do formulário em tempo de execução.

**Formato de número**

Uma sequência de caracteres que define o formato da entrada numérica no campo. Por exemplo, digite \$#.## para exibir \$3.14.

**Orientação**

Especifica se os grupos de opções e as caixas de seleção devem ser organizados horizontal ou verticalmente no formulário. O padrão é a orientação vertical.

**Tamanho da página**

O número de linhas a serem exibidas em cada página de uma tabela ou matriz simples.

**Senha**

Um valor booleano (verdadeiro ou falso) para um campo de texto, geralmente usado em campos de senha. Quando verdadeiro, a entrada do usuário é exibida como caracteres com marcadores para ocultar a entrada de outros usuários. Quando falso ou em branco, a entrada do usuário é exibida exatamente como digitada.

**Padrão**

Uma restrição de entrada ou um requisito de validação para os valores inseridos por um usuário em um campo ou uma área de texto. Por exemplo, defina Padrão como [a-z] para exigir apenas caracteres alfabéticos minúsculos no campo associado.

**Mensagem de padrão**

A dica ou o alerta na tela a ser exibido quando uma entrada do usuário não corresponde ao padrão definido na propriedade Padrão.

### **Processo**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Processo. Quando falso ou em branco, os objetos de automação do tipo Processo não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### **Exibição de processos**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Exibição de processos. Quando falso ou em branco, os objetos de automação do tipo Exibição de processos não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### **Processar como HTML**

Um valor booleano (verdadeiro ou falso) para os campos de texto de várias linhas. Quando verdadeiro, o formulário interpreta o conteúdo do campo como código HTML e o produto exibe o campo da mesma maneira que um navegador da web. Por exemplo, o texto marcado com <H2> é exibido como um cabeçalho de segundo nível, e o marcado com <b> é exibido em negrito.

**Obrigatório**

Um valor booleano (verdadeiro ou falso) que indica se o elemento do formulário deve conter um valor ou pode permanecer em branco.

**Recurso**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Recurso. Quando falso ou em branco, os objetos de automação do tipo Recurso não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

**Cronograma**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Programação. Quando falso ou em branco, os objetos de automação do tipo Programação não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

**Formulário de solicitação inicial**

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os objetos de automação do tipo Formulário de solicitação inicial. Quando falso ou em branco, os objetos de automação do tipo Formulário de solicitação inicial não são exibidos. As propriedades do objeto dos campos de referência de objeto filtram tipos específicos de objetos de automação quando o usuário exibe as opções disponíveis.

### Estilo

Uma ou mais instruções (conforme os exemplos a seguir) que determinam como os atributos de texto em um campo são exibidos aos usuários. Atribuições de estilo de formatação com maiúsculas e minúsculas.

```
color:blue;
textAlign:center;
textDecoration:underline;
textTransform:uppercase
textIndent:30px;
fontStyle:italic;
fontFamily:"Courier";
fontSize:14px;
```

### Índice de guias

Um número que controla a ordem de tabulação do formulário em tempo de criação e de execução. O aplicativo não exige exclusividade. Por exemplo, você pode usar múltiplos de 5 ou 10 ao definir a ordem de tabulação. Entretanto, se inserir um novo campo no layout entre o campo 20 e o campo 25, você poderá atribuir o índice de tabulação como 22.

Você também pode deixar o valor Tab Index em branco e clicar em Mover para cima ou Mover para baixo na barra de ferramentas para ajustar a ordem de tabulação. Por padrão, a ordem de tabulação do formulário segue o layout, da parte superior para a inferior.

### Alinhamento do texto

Especifica como um arquivo de imagem é exibido na parte do layout do formulário que ele ocupa. As imagens podem ser alinhados à esquerda ou à direita, centralizadas ou justificadas.

### Direção do texto

Especifica como são exibidos os caracteres em um campo relativo às bordas esquerda e direita do campo. Selecione **ltr** ou deixe a propriedade em branco para exibir o texto da esquerda para a direita. ltr é o valor padrão e a direção de texto padrão para conjuntos de código ISO-8859 Latin I. Selecione **rtl** para exibir o texto da direita para a esquerda.

### Touchpoint

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os tipos de objetos de automação definidos como verdadeiro. O usuário pode procurar touchpoints por domínio, ambiente e orquestrador. Quando falso ou em branco, somente os objetos do touchpoint padrão são exibidos e o usuário não pode selecionar outro touchpoint.

### Grupo de touchpoints

Um valor booleano (verdadeiro ou falso) para campos de referência ao objeto. Quando verdadeiro, a caixa de diálogo de pesquisa Navegador de objetos inclui os tipos de objetos de automação definidos como verdadeiro. O usuário pode procurar grupos de touchpoints por domínio, ambiente e orquestrador. Quando falso ou em branco, somente os objetos do grupo de touchpoints padrão são exibidos e o usuário não pode selecionar outro grupo de touchpoints.

### URL

Em elementos de formulário Imagem, o caminho do URL para um arquivo de imagem, conforme os exemplos a seguir:

`https://www.<company_name>.com/images/logo1.png`

ou

`http://my.intranet.site/Corporate%20Images/Big2.jpg`

Use imagens para exibir dados, indicar o status, ou fornecer botões nos quais os usuários podem clicar.

### Valor

Os dados armazenados para um elemento de formulário. Todos os elementos em um formulário armazenam seus valores diretamente em suas bordas visuais temporariamente antes de confirmá-las para o banco de dados. O valor é exibido como uma propriedade para dois elementos do formulário específicos:

- Cada opção para um campo de seleção tem sua própria propriedade Valor. Use a propriedade para armazenar uma sequência de caracteres ou um número para cada opção. Por exemplo, embora o usuário veja as opções Fácil, Moderado e Difícil, as propriedades de Valor para essas opções podem ser 0,8, 1,0 e 1,2. Esses valores podem ser usados em cálculos adicionais.
- Um elemento do formulário HTML mostra uma propriedade Valor. Use a propriedade para especificar o texto e o código reais a serem armazenados no campo. Por exemplo, insira os seguintes dados na propriedade Valor para um elemento do formulário HTML:

```
1, 2, 3, <b>4</b>, 5, <i>6</i>, ... <H2>CA Process Automation</H2>
```

O usuário do formulário vê os seguintes dados processados no mesmo campo:

```
1, 2, 3, 4, 5, 6, ...
```

**CA Process Automation**

### Largura

A quantidade de espaço horizontal (em pixels) que o elemento do formulário deve ocupar. Especifique um valor para essa propriedade a partir da borda esquerda do elemento do formulário.

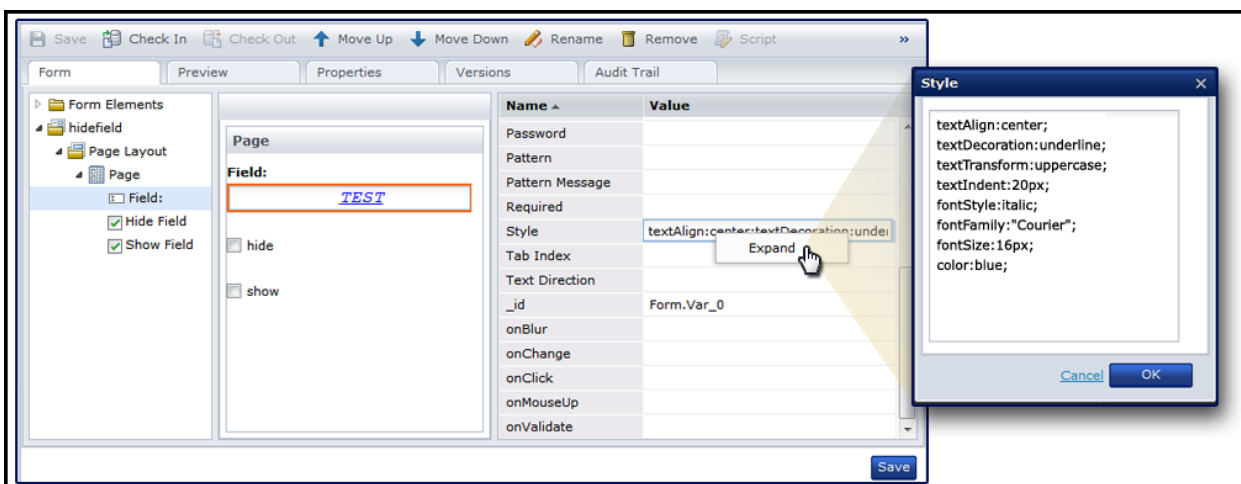
## Exibir propriedades do formulário em uma janela separada

Quando as propriedades do formulário não estiverem visíveis no painel Propriedades do criador de formulários, será possível exibi-las em uma janela separada.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Abra um formulário.
3. No criador de formulário.
  - a. Arraste um campo para uma página no layout.
  - b. Clique no campo listado na estrutura do formulário.
  - c. Identifique uma propriedade ou evento que possa armazenar várias linhas de texto, como Style ou onMouseOver.
  - d. Clique com o botão direito do mouse no valor da propriedade ou do evento e clique em Expandir.

As propriedades expandidas são exibidas em uma janela separada.



**Observação:** ao expandir um evento, é possível pressionar Ctrl + Espaço para exibir sugestões de assistência de conclusão de código. Pressione Ctrl + Alt para exibir uma lista das funções.

## Eventos de elemento do formulário

### onBlur

Ocorre quando um elemento do formulário perde o foco. Por exemplo, um formulário contém um campo Nome de usuário. O usuário estabelece o foco no campo pressionando a tecla Tab até alcançá-lo ou clicando nele. O evento onBlur ocorre quando o usuário realiza uma das seguintes ações:

- O usuário clica em outro objeto ou em outra janela.
- O usuário pressiona a tecla Tab para ir para o próximo campo (por exemplo, o campo Senha).

### onChange

Ocorre quando um elemento de formulário perde o foco, e o novo valor de um elemento de formulário é diferente do valor antigo. Por exemplo, um formulário contém um campo chamado Quantidade com um valor igual a 10. Depois que o usuário alterar o valor para 15, ele não moverá o foco para outro campo. O evento onChange ocorre quando somente o usuário realiza uma das seguintes ações:

- O usuário clica em outro objeto ou em outra janela.
- O usuário pressiona a tecla Tab para ir para o próximo campo.

### onClick

Ocorre quando um usuário clica em um elemento do formulário. Um clique válido inclui os eventos onMouseDown e onMouseUp no mesmo objeto. Esse requisito ajuda a evitar as funções de chamada ou outro código acidentalmente, já que o mouse deve permanecer no objeto clicável.

### onFocus

Ocorre quando um elemento do formulário recebe o foco. Para estabelecer o foco em um elemento do formulário, o usuário pressiona a tecla Tab até alcançá-lo ou clica nele. Também é possível gravar scripts ou código para estabelecer o foco em um elemento do formulário.

### onKeyDown

Ocorre quando um usuário pressiona uma tecla para baixo (por exemplo, quando um usuário pressiona a tecla Tab até o campo Nome ou clica nele). Um script associado ao evento onKeyDown do campo Nome alerta os usuários quando eles tentam usar teclas numéricas.

### onKeyPress

Ocorre após um usuário pressionar uma tecla e mantê-la pressionada. Por exemplo, o evento onKeyPress ocorrerá depois que um usuário pressionar a tecla Tab em um campo Seleccionar chamado Nome ou clicar nele e pressionar alguma tecla que represente um alfabeto. Um script associado ao evento onKeyPress do campo Nome percorre nomes que correspondem à tecla alfabética pressionada.

### onKeyUp

Ocorre quando o usuário libera uma tecla depois de pressioná-la. Por exemplo, o evento `onKeyUp` ocorre quando um usuário pressiona a tecla Tab até um campo Controle giratório chamada Temperatura ou clica nele. Um script associado ao evento aumenta ou diminui o valor de um campo sempre que o usuário pressiona e solta uma tecla específica.

#### **onLoad**

Ocorre quando o formulário é aberto pela primeira vez que o usuário o preencha.

#### **onLookup**

Ocorre quando um usuário clica em Procurar em um campo Pesquisa para exibir os valores que são calculados (ou "pesquisados" conforme instruído pelo script) com base em outros valores de campo.

#### **onMouseDown**

Ocorre quando um usuário pressiona o botão esquerdo do mouse em um elemento do formulário.

#### **onMouseMove**

Ocorre quando um usuário move o ponteiro do mouse dentro do limite de um elemento do formulário.

#### **onMouseOut**

Ocorre quando um usuário move o ponteiro do mouse fora do limite de um elemento do formulário.

#### **onMouseOver**

Ocorre quando o usuário move o ponteiro do mouse sobre um elemento do formulário, e o usuário para de mover o mouse.

#### **onMouseUp**

Ocorre quando o usuário libera o botão esquerdo do mouse, depois de pressioná-lo para baixo em um elemento do formulário.

#### **onMouseWheel**

Ocorre quando um usuário com um mouse equipado com roda rola para frente ou para trás de um elemento do formulário.

#### **onSubmit**

Ocorre quando o usuário envia o formulário. Qualquer uma das seguintes ações pode enviar o formulário:

- O usuário clica em um botão Enviar.
- O usuário pressiona uma tecla específica ou uma combinação de teclas.
- O valor de um campo é alterado.
- O usuário chega ao fim do formulário na ordem de tabulação.

#### **onValidate**

Ocorre quando o código associado verifica o valor de um campo com base nas regras comerciais antes de uma das seguintes ações ocorrer:

- O valor do campo é armazenado no buffer de dados.
- O valor do campo é gravado no banco de dados.

Por exemplo, o usuário pressiona a tecla Tab até o campo Número de série que deve começar com as letras SN e conter 10 dígitos numéricos ou clica nele. Antes que o usuário possa usar a tecla Tab para ir para o próximo campo ou clicar fora do campo, o evento `onValidate` e seu código associado verificam os dados. Você pode alertar o usuário se o número de série não estiver de acordo com as regras de validação para que ele possa ajustar a entrada.

Você pode usar `onValidate` para validação personalizada do campo de entrada. Por exemplo, para garantir que uma entrada do campo tenha pelo menos três caracteres, é possível escrever uma função personalizada na caixa de diálogo Script:

```
validateValue:function(_val) {  
    if(_val.length < 3) {  
        return "Digite mais de 3 caracteres para este campo";  
    } else {  
        return null;  
    }  
}
```

No Criador de formulários, inclua o valor de atributo `onValidate` no campo de texto em que deseja executar a validação. Por exemplo:

```
ca_fd.js.validateValue(_val)
```

A função personalizada substitui o parâmetro obrigatório `_val` pelo valor do campo correto quando o script é executado.

Se o script de validação retornar um valor nulo, a entrada do campo passa a validação. Caso contrário, a entrada do campo falha ao validar e o script retorna um erro (por exemplo, "Digite mais de três caracteres para este campo").

## Funções de elemento do formulário

Muitos eventos da interface do usuário ocorrem quando um usuário interage com um formulário. Os eventos, como alterar o valor de um campo (onChange) ou posicionar o cursor do mouse sobre uma tabela (onMouseOver), podem iniciar funções JavaScript. Como um criador de formulários, você pode usar as funções do elemento do formulário para obter estes e outros objetivos:

- Controle dinamicamente os elementos do formulário com base nos eventos.
- Obter dados de uma tabela ou campo.
- Definir os dados em uma tabela ou campo.
- Usar lógica para ajudar os usuários na navegação por opções potencialmente complexas. Por exemplo, você pode usar as funções para atualizar uma lista de cidades de acordo com o país selecionado pelo usuário.
- Ajudar os usuários a diminuir os erros de entrada de dados.

Esta seção lista as funções JavaScript predefinidas que podem ser usadas nos campos de formulários de solicitação inicial e de interação.

## Elementos do formulário

Elementos do formulário especificam as informações em um formulário que deve ser enviado a um site ou um serviço. Um elemento pode ser do tipo campo de texto, caixa de seleção, senha, botão de opção, botão de envio, etc.

Os seguintes elementos do formulário estão disponíveis no Criador de formulários.

## Funções gerais para todos os elementos do formulário

Essas funções se aplicam a todos os tipos de elementos.

## ca\_pam\_disableField(\_id, isDisable)

Ativa ou desativa um campo especificado. Quando um campo estiver ativado, os usuários poderão fornecer entrada ou editar os valores. Campos desativados ainda estão visíveis, mas não aceitam entrada do usuário.

### Parâmetros de entrada

#### *\_id* (string)

Especifica a identificação exclusiva do campo para ativar ou desativar. Você pode localizar essa identificação no elemento de propriedades. Consulte [Propriedades do elemento do formulário](#) (na página 284) para obter mais informações.

#### *isDisable* (boolean)

Especifica se o campo está desativado (true) ou não (false).

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo desativa o campo da lista de verificação especificado, incluindo todos os itens da lista de verificação nele.

```
ca_pam_disableField('Form.ckListField27', true)
```

## ca\_pam\_hideField(\_id, isHide)

Mostra ou oculta um campo especificado. Os campos ocultos não são visíveis para o usuário. A ordem de tabulação do formulário ignora os campos ocultos. Embora o campo oculto ainda possa conter um valor, ele é efetivamente removido do formulário.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para exibir ou ocultar.

#### ***ocultar*** (booleano)

Especifica se o campo está oculto (true) ou exibido (false).

### Valor de retorno

Nenhum.

### Exemplo

Considere que um usuário preenche um formulário de inscrição online com detalhes sobre o local de nascimento. Se as cidades listadas no campo Cidade de nascimento forem apenas capitais, o usuário poderá selecionar Outra cidade. Ao clicar em Outra cidade, o atributo Oculto será definido como falso e um campo de texto ficará visível. Agora, o usuário poderá digitar o nome da cidade.

Este exemplo oculta o campo especificado.

```
ca_pam_hideField('Form.Var_0', true)
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 02 Mostrar e ocultar elemento do formulário: Mostrar e ocultar elemento do formulário.

## Análise de XML

### ca\_pam\_showDataInTable(result, \_id, tableHeader)

Exibe as linhas de dados resultantes de uma função de campo Pesquisar como uma tabela de coluna única que tenha o cabeçalho ou o título especificado.

Esta função só exibe o "resultado" passado para ela. O resultado aqui representa uma matriz de linhas. O resultado pode ser armazenado em uma variável e passado para esta função ou substituído diretamente com a função que retorna o resultado.

#### Parâmetros de entrada

##### **result (object)**

Especifica uma chamada de função que resulta nos valores a serem exibidos em uma tabela.

##### **\_id (string)**

Define a identificação exclusiva da tabela Campo de pesquisa na qual mostrar o conjunto de resultados.

##### **tableHeader (string)**

Especifica uma sequência de caracteres de texto a ser exibida no início da tabela Campo de pesquisa no formulário.

#### Valor de retorno

Nenhum valor é retornado; essa função só exibe uma tabela com os valores incluídos nela.

#### Exemplo

```
ca_pam_showDataInTable(result, 'Form.office', 'Office');
```

Esse exemplo mostra uma tabela chamada Office que lista os resultados de uma consulta separada para obter nomes de escritórios. Na consulta separada, a seleção de região do usuário determina os escritórios a serem exibidos.

O evento onLookup para um campo de pesquisa chamado Escritório chama esta função.

## ca\_pam\_convertToSimpleArray (objectArray, fieldName)

Cria uma matriz simples de sequências de caracteres a partir de qualquer matriz de objetos JavaScript mais complexa.

### Parâmetros de entrada

objectArray (object)

Especifica uma referência a uma matriz de objetos JavaScript que contém várias propriedades.

fieldName (string)

Define o nome da propriedade no objectArray a partir do qual criar a matriz simples.

### Valor de retorno

A matriz que o método cria é retornado.

### Exemplo

```
var location1 = new Object();
location1.buildingCode = 10;
location1.floorCode = 20;
var location2 = new Object();
location2.buildingCode = 100;
location2.floorCode = 200;
var LocationArray = {location1,location2};

var floorCodeArray = ca_pam_convertToSimpleArray (LocationArray,
'floorCode');
```

A saída de floorCodeArray {20,200}.

## ca\_pam\_convertToJavaScriptObject(valueObject)

Converte um tipo de dados de automação de processos em um objeto JavaScript padrão de acordo com o tipo de dados de entrada. Se você fornecer uma sequência de caracteres como um valor de entrada, o método retornará um valor de sequência de caracteres.

### Parâmetros de entrada

#### valueObject

Especifica os tipos de dados de automação de processos. Por exemplo, Booleano, Data e ValueMap.

### Valor de retorno

Retorna um objeto JavaScript com base no valor de valueObject. Se ValueMap for o valor de entrada, ele retornará um objeto JavaScript como um mapa. Se a matriz Valor for o parâmetro de entrada, o método retornará uma matriz JavaScript.

**Observação:** para o tipo de dados Data, o valor longo é retornado, que é a representação canônica da data. Para convertê-lo em um objeto de data, use a seguinte sintaxe:

```
new Date (ca_pam_convertToJavaScriptObject(valueObject))
```

### Exemplo

Este exemplo declara uma variável (array) e a define como o objeto JavaScript convertido.

```
var array = ca_pam_convertToJavaScriptObject(valueObject);
```

## Caixa de seleção

### ca\_pam\_isSelectedCheckBox(\_id)

Determina se uma caixa de seleção está marcada (verdadeiro) ou desmarcada (falso).

#### Parâmetros de entrada

##### ***\_id*** (string)

Define a identificação exclusiva do campo Caixa de seleção a ser avaliado.

#### Valor de retorno

##### ***Value*** (boolean)

True se a caixa de seleção estiver selecionada, caso contrário, retorna false.

#### Exemplo

```
ca_pam_isSelectedCheckBox( ' Form.ckbxInsBuy' )
```

O rótulo da caixa de seleção ckbxInsBuy é Comprar seguro. Este código de exemplo retorna "verdadeiro" quando a caixa de seleção Comprar seguro está marcada. O código retorna "falso" quando a caixa de seleção está desmarcada.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 02 Mostrar e ocultar elemento do formulário: Mostrar e ocultar elemento do formulário.

## ca\_pam\_selectCheckBox(\_id, isSelect)

Marca ou desmarca uma caixa de seleção especificada.

### Parâmetros de entrada

#### **\_id (string)**

Especifica a identificação exclusiva do campo Caixa de seleção a ser selecionado ou desmarcado.

#### **isSelect (boolean)**

Especifica se um campo está selecionado (true) ou desmarcado (false).

### Valor de retorno

Nenhum.

### Exemplo

```
ca_pam_selectCheckBox('Form.ckbxInsBuy', true)
```

Este exemplo marca a caixa de seleção *ckbxInsBuy* no formulário.

- O rótulo da caixa de seleção *ckbxInsBuy* é Comprar seguro.
- A função `ca_pam_selectCheckBox` é útil se o usuário selecionar outras opções no formulário que exijam a compra de seguro.

## Data

Exceto conforme indicado nesta seção, todas as funções de data retornam a data como uma sequência de caracteres no formato especificado na propriedade Formato de data.

## ca\_pam\_getDateFieldMaxValue(\_id)

Retorna o valor da propriedade Valor máximo no campo Data como uma sequência de caracteres.

### Parâmetros de entrada

#### **\_id (string)**

A identificação exclusiva de um campo Data.

### Valor de retorno

O valor máximo de um campo Data como uma sequência de caracteres.

### Exemplo

Este exemplo retorna o valor máximo permitido para o campo *DateDeparture* como uma sequência de caracteres de data (por exemplo, 20/05/2025).

```
ca_pam_getDateFieldMaxValue('Form.DateDeparture')
```

## ca\_pam\_getDateFieldMaxValueInMillis(\_id)

Retorna o valor da propriedade Valor máximo no campo Data, expresso como um número inteiro longo. Esse número inteiro representa o número de milissegundos antes ou depois de 1º de janeiro de 1970 00:00:00 UTC (conhecido como *Unix Epoch*). É possível passar o valor nesse campo, conforme mencionado no método [ca\\_pam\\_getDateFieldMaxValue \( \\_id\)](#) (na página 307).

### Parâmetros de entrada

#### **`_id` (string)**

Define o identificador exclusivo de um campo de data.

### Valor de retorno

O valor máximo de um campo Data, expresso como um número inteiro longo.

### Exemplo

```
ca_pam_getDateFieldMaxValueInMillis( 'Form.DateDeparture' )
```

Este exemplo retorna o valor máximo permitido para o campo *DateDeparture* como uma sequência de caracteres numéricos medidos em milissegundos. Para um campo de data com um valor máximo de 25/05/2025, essa função retornaria 1748188800000 (isto é, 1.748.188.800.000 ou 1,7 trilhões de milissegundos).

A função `ca_pam_getDateFieldMaxValueInMillis` retorna um resultado negativo para datas anteriores a 1970.

## ca\_pam\_getDateFieldMinValue(\_id)

Retorna o valor da propriedade Valor mínimo no campo Data como uma sequência de caracteres. É possível passar o valor nesse campo, conforme mencionado no método [ca\\_pam\\_getDateFieldMaxValue \( \\_id\)](#) (na página 307).

### Parâmetros de entrada

#### **`_id` (string)**

Especifica a identificação exclusiva de um campo Data.

### Valor de retorno

O valor mínimo de um campo Data como uma sequência de caracteres.

### Exemplo

Este exemplo retorna o valor mínimo permitido para o campo *DateDeparture* como uma sequência de caracteres de data (por exemplo, 05/05/2025).

```
ca_pam_getDateFieldMinValue( 'Form.DateDeparture' )
```

## ca\_pam\_getDateFieldMinValueInMillis(\_id)

Retorna o valor da propriedade Valor mínimo no campo Data, expresso como um número inteiro longo. Esse número inteiro representa o número de milissegundos antes ou depois de 1º de janeiro de 1970 00:00:00 UTC (conhecido como *Unix Epoch*). É possível passar o valor nesse campo, conforme mencionado no método [ca\\_pam\\_getDateFieldMaxValue \( id\)](#) (na página 307).

### Parâmetros de entrada

#### **`_id` (string)**

Especifica a identificação exclusiva de um campo Data.

### Valor de retorno

O valor mínimo de um campo Data, expresso como um número inteiro longo.

### Exemplo

```
ca_pam_getDateFieldMinValueInMillis('Form.DateOfBirth')
```

Este exemplo retorna o valor mínimo permitido para o campo DateOfBirth como uma sequência de caracteres numéricos medidos em milissegundos. Para um campo de data com um valor mínimo de 01/01/1974, essa função retornaria 126291600000 (isto é, 126.291.600.000 ou 126 bilhões de milissegundos).

Essa função retorna um resultado negativo para datas antes de 1970.

## ca\_pam\_getDateFieldValue(\_id)

Recupera o valor de um campo Data como uma sequência de caracteres de data (por exemplo, 05/05/2025).

### Parâmetros de entrada

#### **`_id` (string)**

Especifica a identificação exclusiva de um campo Data.

### Valor de retorno

O valor da data como uma sequência de caracteres.

### Exemplo

```
var LastDay =ca_pam_getDateFieldValue('Form.TripEndDate');  
ca_pam_setDateFieldMaxValue('Form.DateDeparture',LastDay);
```

A primeira linha deste exemplo pega a TripEndDate (por exemplo, 15/05/2014) e a armazena na variável LastDay.

A segunda linha deste exemplo define o valor máximo do campo DateDeparture para o valor da variável LastDay.

## ca\_pam\_getDateFieldValueInMillis(\_id)

Recupera o valor de um campo Data como um número inteiro longo (por exemplo, 61238000). Esse número inteiro representa o número de milissegundos antes ou depois de 1º de janeiro de 1970 00:00:00 UTC (conhecido como *Unix Epoch*).

### Parâmetros de entrada

#### ***\_id*** (string)

Define o identificador exclusivo de um campo de data.

### Valor de retorno

Valor de data para um campo Data que é expresso como um número inteiro longo.

### Exemplo

```
ca_pam_getDateFieldValueInMillis('Form.DateofBirth')
```

Este exemplo retorna o valor para o campo DateofBirth como uma sequência de caracteres numéricos medidos em milissegundos. Para um campo de data com um valor de 22/08/1965, essa função retornaria -132307200000 (ou seja, -132.307.200.000 ou -132 bilhões de milissegundos).

A função ca\_pam\_getDateFieldValueInMillis retorna um resultado positivo para as datas posteriores a 1º de Janeiro de 1970.

## ca\_pam\_setDateFieldMaxValue(\_id, val)

Define o valor da propriedade Valor máximo do campo Data. Os usuários do formulário não podem inserir uma data no campo de data que seja posterior ao valor máximo.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Data.

#### ***val***

Especifica o valor a ser usado como um máximo, expresso como uma sequência de caracteres de data (por exemplo, 20/05/2014).

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define a propriedade Valor máximo do campo DateDeparture para a data armazenada na variável LastDay.

```
ca_pam_setDateFieldMaxValue('Form.DateDeparture', LastDay);
```

## ca\_pam\_setDateFieldMinValue(\_id, val)

Define o valor da propriedade Valor mínimo do campo Data. Os usuários do formulário não podem inserir uma data no campo de data que seja anterior ao valor mínimo.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Data.

#### ***val***

Especifica o valor a ser usado como um mínimo, expresso como uma sequência de caracteres de data (por exemplo, 05/05/2012).

### Valor de retorno

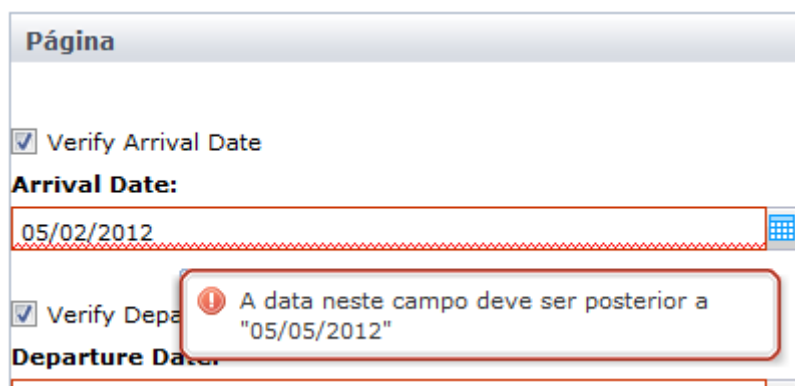
Nenhum.

### Exemplo

```
ca_pam_setDateFieldMinValue('Form.DateArrival', today);
```

Este exemplo define a propriedade Valor mínimo do campo DateArrival para a data armazenada na variável hoje.

- O produto alerta os usuários do formulário caso eles insiram uma data anterior ao valor mínimo.



The screenshot shows a web form titled "Página". It contains two checked checkboxes: "Verify Arrival Date" and "Verify Depa...". Below the first checkbox is the label "Arrival Date:" followed by a date input field containing "05/02/2012". Below the second checkbox is the label "Departure Date:". A red-bordered error message box is overlaid on the "Departure Date" field, containing a red exclamation mark icon and the text: "A data neste campo deve ser posterior a '05/05/2012'".

- Os calendários de pop-up no formulário exibem as datas inválidas em cinza e não permitem que os usuários cliquem nas datas inválidas.

## ca\_pam\_setDateFieldValue(\_id, val)

Define o valor de um campo de data como uma sequência de caracteres de data (por exemplo, 15/05/2014). O valor é definido apenas se ele for especificado no formato correto (ou seja, no formato especificado na propriedade Formato de data do campo Data). Se o valor dessa propriedade estiver em branco, então o valor será definido no Formato de data especificado nas configurações do usuário.

**Observação:** passe a data no formato correto; caso contrário, ela não será configurada.

### Parâmetros de entrada

**\_id**

Define o identificador exclusivo de um campo de data.

**val**

Define um valor de data que você especificar (por exemplo, 05/05/2014).

### Valor de retorno

Nenhum.

### Exemplo

```
ca_pam_setDateFieldValue('Form.DateArrival', '05/05/2014');
```

Este exemplo define o campo DateArrival para 05/05/2014.

## Campo de pesquisa

### ca\_pam\_getTextFieldValue(\_id)

Retorna um valor a partir de um campo Texto.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Texto.

#### Valor de retorno

O *valor* do campo Texto (string).

#### Exemplo

Este exemplo retorna o valor do campo Texto de top\_vendors.

```
ca_pam_getTextFieldValue(' Form.top_vendors')
```

Este exemplo retorna o valor do campo Texto de top\_vendors e define o campo Texto de dynamic\_field como o valor retornado.

```
ca_pam_setTextFieldValue(' Form.dynamic_field',  
ca_pam_getTextFieldValue(' Form.top_vendors'))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_setTextFieldValue(\_id, val)

Define o valor de um campo Texto.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual definir um valor específico.

#### ***val*** (string)

Define o novo valor para o campo Texto.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define o campo de texto top\_vendors como "Eu voto no fornecedor 3 porque:"

```
ca_pam_setTextFieldValue('Form.top_vendors', 'I vote for vendor 3 because: ')
```

Para continuar este exemplo, use ca\_pam\_getTextFieldValue('Form.top\_vendors') depois que os usuários inserirem seus motivos para enviar suas declarações completas.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Este exemplo retorna o valor do campo top\_vendors e define o dynamic\_field como este valor.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Grupo de botões de opção

### ca\_pam\_isSelectRadio (\_id, radio\_id)

Determina se um botão de opção específico está selecionado.

#### Parâmetros de entrada

##### ***\_id*** (string)

Define a identificação exclusiva de um campo Grupo de botões de opção.

##### ***radio\_id*** (string)

Especifica a identificação exclusiva de um botão de opção específico em um grupo.

#### Valor de retorno

##### ***Value*** (boolean)

True se a caixa de seleção estiver selecionada; caso contrário, retorna false.

#### Exemplo

```
ca_pam_isSelectRadio('Form.Transportation',  
'Form.Transportation.Option_01_Air')
```

*Transporte* é um grupo com dois botões de opção com as seguintes IDs:

- Option\_01\_Air
- Option\_02\_Rail

Esse exemplo retorna "true", se o botão de opção *Aéreo* no grupo *Transporte* estiver selecionado ou "false" se nenhuma outra opção estiver selecionada.

## **ca\_pam\_selectRadio(\_id, radio\_id)**

Seleciona um botão de opção específico em um campo Grupo de botões de opção.

### **Parâmetros de entrada**

#### ***\_id* (string)**

Define a identificação exclusiva de um campo Grupo de botões de opção.

#### ***radio\_id* (string)**

Especifica a identificação exclusiva de um botão de opção em um Grupo de botões de opção.

### **Valor de retorno**

Nenhum.

### **Exemplo**

```
ca_pam_selectRadio('Form.Transportation',  
'Form.Transportation.Option_01_Air')
```

Este exemplo seleciona o botão de opção *Air* no grupo de botões de opção *Transporte*.

*Transporte* é um grupo de botões de opção com dois botões de opção com as seguintes IDs:

- Option\_01\_Air
- Option\_02\_Rail

## Selecionar

### ca\_pam\_getSelectedOptions(\_id)

Retorna o valor de índice da opção selecionada atualmente em um campo Selecionar.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva do campo Selecionar para o qual recuperar o valor da opção.

#### Valor de retorno

Essa função retorna uma matriz de índice de opções selecionadas.

No caso de uma única lista suspensa Selecionar, o índice da opção selecionada pode ser recuperado, acessando o primeiro elemento da matriz.

#### Exemplo

```
ca_pam_getSelectedOptions('Form.VarFillerType')
```

Este exemplo recupera o valor de índice do Filler Type (tipo de preenchimento) que o usuário escolheu.

- *Filler Type* é um campo de seleção com opções de Rocks [0], Sand [1] e Water [2].
- A função `ca_pam_getSelectedOptions` retorna o valor de índice 1 quando o usuário seleciona Sand na lista suspensa Filler Type.

## ca\_pam\_getSelectedOptionValues(\_id)

Retorna o valor da opção selecionada atualmente em um campo Selecionar.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual recuperar o valor da opção.

### Valor de retorno

Essa função retorna uma matriz de valor de opção selecionada.

No caso de uma única lista suspensa Selecionar, o valor da opção selecionada pode ser recuperado, acessando o primeiro elemento da matriz.

### Exemplo

```
alert(ca_pam_getSelectedOptionValues('Form.VarFillerType')[0])
```

Este exemplo retorna o número de valor ou sequência de caracteres de texto da opção para o Filler Type (tipo de preenchimento) que o usuário escolheu.

- O comando de alerta exibe o resultado do valor de opção para o usuário em uma caixa de diálogo pop-up ou em uma mensagem.
- *Filler Type* é um campo de seleção com as opções, valores de índice mostrados entre colchetes e valores abaixo:

Rocks [0], \$50

Sand [1], \$100

Water [2], chamada para obter detalhes.

- A função `ca_pam_getSelectedOptionValues` retorna o valor da opção selecionada na lista suspensa Filler Type (Tipo de preenchimento) quando o usuário clica em Atualizar formulário de atualização.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 06 Preencher BD do menu suspenso: Preencher BD do menu suspenso.

## ca\_pam\_selectOption(\_id, name, value, isSelect)

Seleciona ou desmarca uma opção em um campo Selecionar com base no nome e no valor da opção. Também é possível usar essa função para impedir a seleção de uma opção específica.

Essa função pode selecionar e desmarcar uma opção. Se uma opção não estiver selecionada e você desejar selecioná-la programaticamente, então `isSelect` será `true`. Se uma opção já estiver selecionada e você desejar desmarcá-la, então use `isSelect` como `false`.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo Selecionar a ser definido.

#### ***name*** (string)

Especifica o nome da opção específica do campo Selecionar.

#### ***value*** (string)

Define o nome da opção específica do campo Selecionar. Esse valor é o valor existente da opção.

#### ***isSelect*** (boolean)

Especifica se o campo está selecionado (`true`) ou desmarcado (`false`).

### Valor de retorno

Nenhum.

### Exemplo

```
ca_pam_selectOption('Form.City', 'Regional Hub', 'Sydney', true);
```

Este exemplo define o campo Cidade para a opção nomeada Regional Hub.

Se várias opções forem nomeadas Regional Hub, então essa função se aplicará à opção com o *valor* especificado. Essa diferença é importante. Você pode utilizar o `ca_pam_getSelectedOptionValues(_id)` para identificar o *valor* específico para uma opção de campo, mesmo quando várias opções tiverem o mesmo nome. Por exemplo, os usuários podem continuar a utilizar um conjunto de formulários padrão para direcionar os negócios para a Central regional, mesmo que o *valor* seja alterado.

## ca\_pam\_selectOptionByIndex(\_id, index, isSelect)

Seleciona ou desmarca uma opção em um campo Selecionar com base no índice.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo Selecionar para selecionar ou desmarcar.

#### ***index*** (integer)

Especifica a identificação fixa para uma opção no campo.

#### ***isSelect*** (boolean)

Especifica se a opção de campo Selecionar está selecionada (true) ou desmarcada (false).

### Valor de retorno

Nenhum.

### Exemplos

```
ca_pam_selectOptionByIndex('Form.VarFillerType', 2, true)
```

Este exemplo define a opção atual de *Filler Type* para o valor na posição de índice 2.

- *Filler Type* é um campo de seleção com opções de Rocks [0], Sand [1] e Water [2].
- *Economy Option* é uma caixa de seleção com esta função em seu evento onClick.
- Quando o usuário marca a caixa de seleção *Economy Option* (Opção de economia), a função `ca_pam_selectOptionByIndex` define o campo *Filler Type* (Tipo de preenchimento) como a opção na posição 2 do índice, *Water* (Água).

Outro exemplo:

```
ca_pam_selectOptionByIndex('Form.VarFillerType', 2, false)
```

Quando *Filler Type* já estiver definido para *Water*, esta função desmarca o campo *Filler Type*.

## ca\_pam\_addValuesInSelectStore(\_id, values)

Adiciona novas opções no campo Selecionar com um conjunto de valores simples que você definiu no código.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Selecionar.

#### ***values*** (objeto de JavaScript)

Especifica uma matriz de objetos em cada objeto tem duas propriedades ("name" e "value") que são necessárias para representar as opções do campo Selecionar. A propriedade "name" de cada objeto é representada como a chave para a opção adicionada e a propriedade "value" deve ser o valor. Se você tiver uma matriz de nomes e valores, então poderia usar também a função `ca_pam_createSelectStore` para criar o objeto "values" (como mostrado no exemplo).

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo usa a seleção do usuário de Oeste ou Norte em um campo de região separado para atualizar os ícones disponíveis no campo Cidade dinamicamente.

```
if ('West'==regionChoice)
  var cityOptionNames =["New York", "Rio De Janeiro", "Mexico City"]
  var cityOptionValues =["West_01", "West_02", "West_03"];
if ('North'==regionChoice)
  var cityOptionNames =["Madrid", "Moscow", "Copenhagen"]
  var cityOptionValues =["North_04", "North_05", "North_06"];
ca_pam_addValuesInSelectStore('Form1.City',
ca_pam_createSelectStore(cityOptionNames ,cityOptionValues ))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para a pasta Formulários de interação do usuário. A função pode ser encontrada nos seguintes formatos:
  - 04 Preencher: Preencher conjunto de dados do menu suspenso: Preencher conjunto de dados do menu suspenso.
  - 06 Preencher BD do menu suspenso: Preencher BD do menu suspenso.

## ca\_pam\_clearSelectStore(\_id)

Limpa todas as opções do campo Selecionar.

### Parâmetros de entrada

#### **`_id` (string)**

Especifica a identificação exclusiva do campo Selecionar a ser desmarcado.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo desmarca todas as opções existentes no campo Cidade do Form1. Isso pode ser usado para reinicializar o campo Selecionar e preencher novas opções usando o `ca_pam_addValuesInSelectStore ()`.

```
ca_pam_clearSelectStore(Form1.City)
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário e, em seguida, 06 Preencher BD do menu suspenso: Preencher BD do menu suspenso.

## `ca_pam_createSelectStore(nameArray, valueArray)`

Usado para criar um conjunto de opções para um campo Selecionar.

### Parâmetros de entrada

#### ***nameArray*** (array)

Define uma matriz de nomes de opções que são exibidas para o usuário.

#### ***valueArray*** (array)

Opcionalmente, defina uma matriz de valores de opções correspondente para cada item na `nameArray`. Se for omitida, `nameArray` será usada para definir nomes e valores.

### Valor de retorno

O objeto de JavaScript que representa as opções de armazenamento e seus respectivos valores.

### Exemplo

Este exemplo define as opções disponíveis para o campo Cidade para os nomes especificados e valores associados.

```
var cityOptionNames = ["Los Angeles", "New York", "New Carolina"];
var cityOptionValues = ["LA", "NY", "NC"];
var newStore = ca_pam_createSelectStore(cityOptionNames
,cityOptionValues )
```

Agora, a variável `newStore` pode ser passada para um elemento Selecionar formulário usando o seguinte método:

```
ca_pam_addValuesInSelectStore('Form.City',newStore);
```

## **ca\_pam\_createSelectStoreFromSQLResult(resultFromSQLQuery, nameColumnID, valueColumnID)**

Cria um conjunto de opções para um campo Selecionar diretamente do resultado de uma consulta SQL.

### **Parâmetros de entrada**

#### ***resultFromSQLQuery* (object)**

Define os dados recuperados por uma instrução SQL.

#### ***nameColumnID* (array)**

Define o nome da coluna no resultado de SQL que é usado como nomes do campo Selecionar.

#### ***valueColumnID* (array)**

Opcionalmente, define o nome da coluna no resultado de SQL que é usado como valores no campo Selecionar. Se for omitida, a coluna que é definida pela nameColumnID será usada para nomes e valores.

### **Valor de retorno**

A matriz que a função cria é retornada.

### **Exemplo**

Este exemplo usa o resultado de uma consulta de uma origem de dados externa para definir as opções para o campo Cidade. A consulta é executada quando o usuário seleciona uma opção no campo Região.

```
ca_pam_addValuesInSelectStore('Form1.City',ca_pam_createSelectStoreFromSQLResult(result,'txtRegion'));
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário e, em seguida, 06 Preencher BD do menu suspenso: Preencher BD do menu suspenso.

## Campo Texto

### ca\_pam\_getTextFieldValue(\_id)

Retorna um valor a partir de um campo Texto.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Texto.

#### Valor de retorno

O *valor* do campo Texto (string).

#### Exemplo

Este exemplo retorna o valor do campo Texto de top\_vendors.

```
ca_pam_getTextFieldValue('Form.top_vendors')
```

Este exemplo retorna o valor do campo Texto de top\_vendors e define o campo Texto de dynamic\_field como o valor retornado.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_setTextFieldValue(\_id, val)

Define o valor de um campo Texto.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual definir um valor específico.

#### ***val*** (string)

Define o novo valor para o campo Texto.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define o campo de texto top\_vendors como "Eu voto no fornecedor 3 porque:"

```
ca_pam_setTextFieldValue('Form.top_vendors', 'I vote for vendor 3 because: ')
```

Para continuar este exemplo, use ca\_pam\_getTextFieldValue('Form.top\_vendors') depois que os usuários inserirem seus motivos para enviar suas declarações completas.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Este exemplo retorna o valor do campo top\_vendors e define o dynamic\_field como este valor.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Área de texto

### ca\_pam\_getTextFieldValue(\_id)

Retorna um valor a partir de um campo Texto.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Texto.

#### Valor de retorno

O *valor* do campo Texto (string).

#### Exemplo

Este exemplo retorna o valor do campo Texto de top\_vendors.

```
ca_pam_getTextFieldValue('Form.top_vendors')
```

Este exemplo retorna o valor do campo Texto de top\_vendors e define o campo Texto de dynamic\_field como o valor retornado.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_setTextFieldValue(\_id, val)

Define o valor de um campo Texto.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual definir um valor específico.

#### ***val*** (string)

Define o novo valor para o campo Texto.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define o campo de texto top\_vendors como "Eu voto no fornecedor 3 porque:"

```
ca_pam_setTextFieldValue('Form.top_vendors', 'I vote for vendor 3 because: ')
```

Para continuar este exemplo, use ca\_pam\_getTextFieldValue('Form.top\_vendors') depois que os usuários inserirem seus motivos para enviar suas declarações completas.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Este exemplo retorna o valor do campo top\_vendors e define o dynamic\_field como este valor.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Texto de várias linhas

### ca\_pam\_getTextFieldValue(\_id)

Retorna um valor a partir de um campo Texto.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Texto.

#### Valor de retorno

O *valor* do campo Texto (string).

#### Exemplo

Este exemplo retorna o valor do campo Texto de top\_vendors.

```
ca_pam_getTextFieldValue('Form.top_vendors')
```

Este exemplo retorna o valor do campo Texto de top\_vendors e define o campo Texto de dynamic\_field como o valor retornado.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_setTextFieldValue(\_id, val)

Define o valor de um campo Texto.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual definir um valor específico.

#### ***val*** (string)

Define o novo valor para o campo Texto.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define o campo de texto top\_vendors como "Eu voto no fornecedor 3 porque:"

```
ca_pam_setTextFieldValue('Form.top_vendors', 'I vote for vendor 3 because: ')
```

Para continuar este exemplo, use ca\_pam\_getTextFieldValue('Form.top\_vendors') depois que os usuários inserirem seus motivos para enviar suas declarações completas.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Este exemplo retorna o valor do campo top\_vendors e define o dynamic\_field como este valor.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Table

### `ca_pam_clearTableData(_id, startIndex, endIndex)`

Exclui uma ou mais linhas de dados de uma tabela.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

##### ***startIndex*** (integer)

Define o índice numérico da primeira linha de dados da tabela a ser excluída.

##### ***endIndex*** (integer)

Especifica o índice numérico da última linha de dados da tabela a ser excluída.

#### Valor de retorno

Nenhum.

#### Exemplo

```
ca_pam_clearTableData('Form.employeeList', 0, ca_pam_getTableRowCount('Form.employeeList') - 1);
```

Este exemplo exclui todas as linhas na tabela employeeList.

- O intervalo a ser excluído começa com a linha 0 e termina com a última linha.
- Para calcular a última linha, a função `ca_pam_clearTableData` obtém a contagem de linhas atual da tabela a partir da função `ca_pam_getTableRowCount` e subtrai 1.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para a pasta Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_getTableData(\_id, startIndex, endIndex)

Retorna os dados da tabela a partir do índice de início até o índice final conforme fornecido no nome do método. Essa função retorna uma matriz de objetos JavaScript em que cada elemento da matriz representa uma linha da tabela. Para recuperar o valor de uma coluna específica, é possível acessar o valor usando a propriedade Nome do objeto JavaScript correspondente à linha.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### ***startIndex*** (integer)

Especifica o índice numérico da primeira linha da tabela a partir da qual os dados devem ser retornados.

#### ***endIndex*** (integer)

Especifica o índice numérico da última linha da tabela a partir da qual os dados devem ser retornados. Se você especificar um índice maior do que a última linha disponível, a função falhará.

### Valor de retorno

Retorna uma matriz que é representada como um objeto de JavaScript.

**Importante:** Depois que o método retornar os dados da tabela, os nomes das variáveis usadas para acessar as colunas devem estar em minúsculas, independentemente de como você definir a propriedade Nome de cada campo da coluna no Criador de formulários.

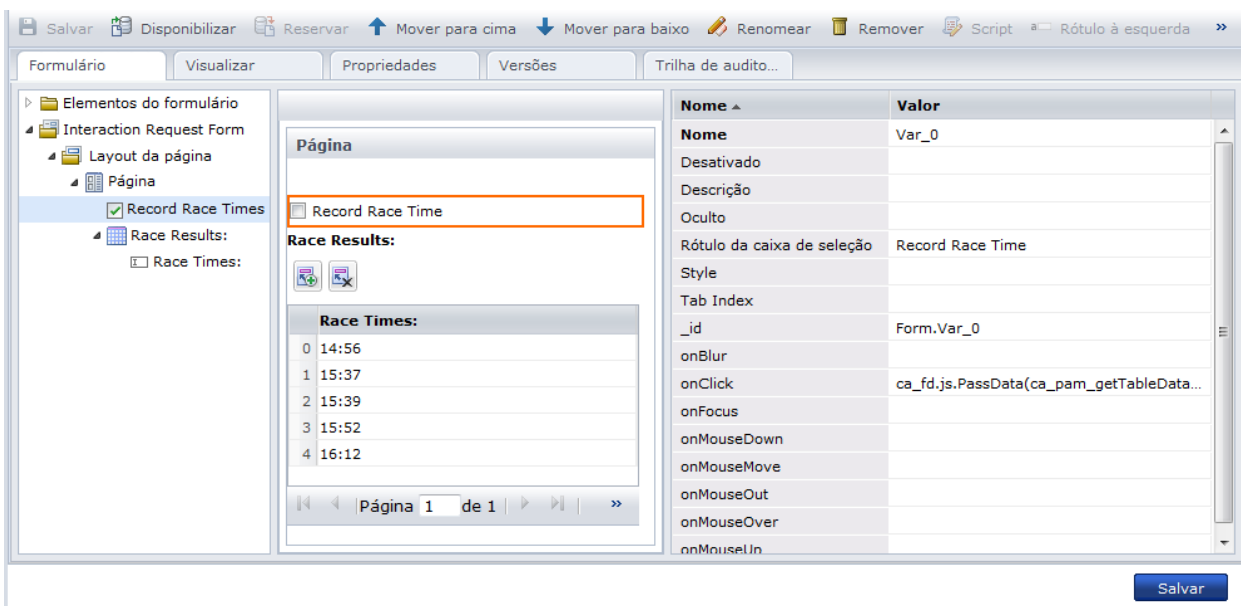
### Exemplo

```
ca_fd.js.PassData(ca_pam_getTableData('Form.tableRaceResults',0,4))
```

Esse exemplo usa o script de formulário a seguir para transmitir os cinco melhores tempos de finalização de uma disputa de um elemento de formulário Tabela para um banco de dados personalizado. O evento onClick da caixa de seleção Record Race Times inicia as chamadas de função para obter os dados da tabela, enviar os dados e registrar os resultados.

```
{
  PassData : function(result) {
    for(i=0;i<result.length;i++)
    {
      var firstRow = result[i];
      alert(firstRow.var_0);
    }
  }
}
```

A ilustração a seguir mostra o formulário como será exibido no momento da criação.



### ca\_pam\_getTableRowCount(\_id)

Retorna o número total de linhas de dados de uma tabela.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### Valor de retorno

Retorna um número de linhas como um número inteiro.

#### Exemplo

```
ca_pam_getTableRowCount('Form.employeeList')
```

Esse exemplo retorna a contagem de linhas do elemento de formulário Tabela da lista de funcionários.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## **ca\_pam\_setTableData(\_id, values)**

Define os dados de uma tabela.

### **Parâmetros de entrada**

#### ***\_id* (string)**

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### ***values* (object)**

Especifica os dados a serem exibidos nas linhas e nas colunas da tabela. Este parâmetro aceita somente o resultado da consulta de SQL.

### **Valor de retorno**

Nenhum.

### **Exemplo**

```
ca_pam_setTableData('Form.employeeList', result);
```

Este exemplo define a tabela employeeList para os valores que a variável de resultado retorna.

## ca\_pam\_setTableDataFromJSObject(\_id, values)

Preenche uma tabela com a matriz de objeto JavaScript.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a ID da tabela.

#### ***values*** (object)

Especifica a matriz do JavaScript que contém uma lista de objetos. As propriedades dos objetos devem ser as mesmas que o nome da coluna para preencher os dados.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo mostra como preencher uma tabela a partir de uma matriz de objeto de JavaScript. A ID da tabela é Form.name e tem duas colunas, firstName e lastName.

```
var tableData = new Array();
var data1 = new Object();
data1.firstName = "firstName1";
data1.lastName = "lastName1";
tableData[0] = data1;
```

```
var data2 = new Object();
data2.firstName = "firstName2";
data2.lastName = "lastName2";
tableData[1] = data2;
ca_pam_setTableDataFromJSObject('Form.names', tableData );
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## Matriz simples

### `ca_pam_clearTableData(_id, startIndex, endIndex)`

Exclui uma ou mais linhas de dados de uma tabela.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

##### ***startIndex*** (integer)

Define o índice numérico da primeira linha de dados da tabela a ser excluída.

##### ***endIndex*** (integer)

Especifica o índice numérico da última linha de dados da tabela a ser excluída.

#### Valor de retorno

Nenhum.

#### Exemplo

```
ca_pam_clearTableData('Form.employeeList',0,ca_pam_getTableRowCount('Form.employeeList')-1);
```

Este exemplo exclui todas as linhas na tabela employeeList.

- O intervalo a ser excluído começa com a linha 0 e termina com a última linha.
- Para calcular a última linha, a função `ca_pam_clearTableData` obtém a contagem de linhas atual da tabela a partir da função `ca_pam_getTableRowCount` e subtrai 1.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para a pasta Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_getTableData(\_id, startIndex, endIndex)

Retorna os dados da tabela a partir do índice de início até o índice final conforme fornecido no nome do método. Essa função retorna uma matriz de objetos JavaScript em que cada elemento da matriz representa uma linha da tabela. Para recuperar o valor de uma coluna específica, é possível acessar o valor usando a propriedade Nome do objeto JavaScript correspondente à linha.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### ***startIndex*** (integer)

Especifica o índice numérico da primeira linha da tabela a partir da qual os dados devem ser retornados.

#### ***endIndex*** (integer)

Especifica o índice numérico da última linha da tabela a partir da qual os dados devem ser retornados. Se você especificar um índice maior do que a última linha disponível, a função falhará.

### Valor de retorno

Retorna uma matriz que é representada como um objeto de JavaScript.

**Importante:** Depois que o método retornar os dados da tabela, os nomes das variáveis usadas para acessar as colunas devem estar em minúsculas, independentemente de como você definir a propriedade Nome de cada campo da coluna no Criador de formulários.

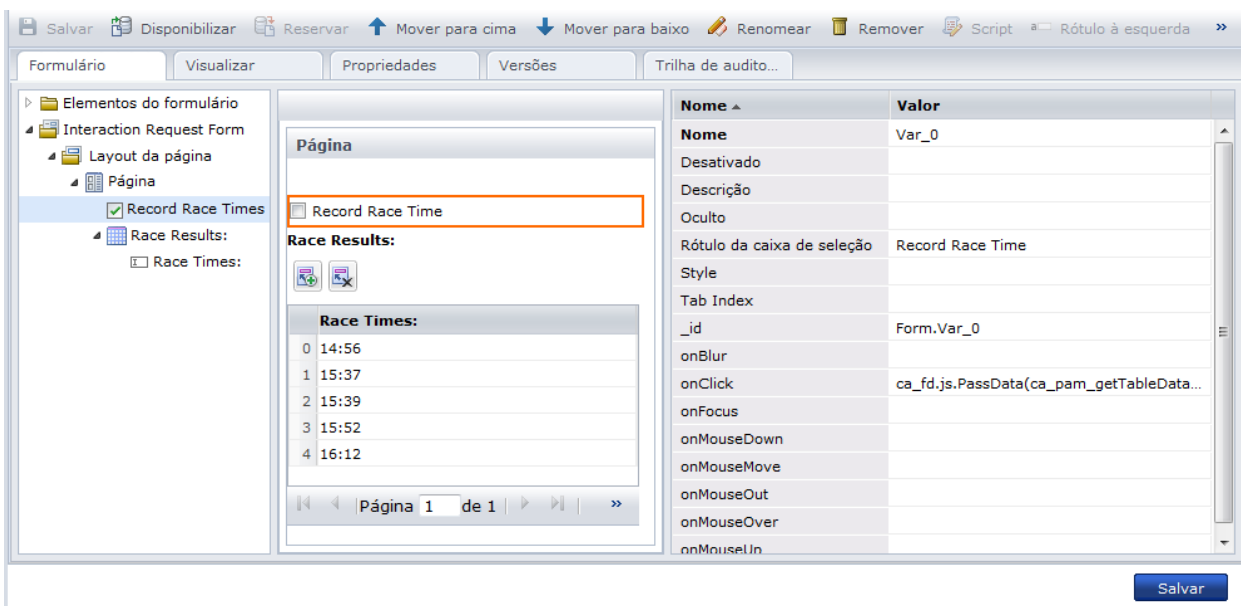
### Exemplo

```
ca_fd.js.PassData(ca_pam_getTableData('Form.tableRaceResults',0,4))
```

Esse exemplo usa o script de formulário a seguir para transmitir os cinco melhores tempos de finalização de uma disputa de um elemento de formulário Tabela para um banco de dados personalizado. O evento onClick da caixa de seleção Record Race Times inicia as chamadas de função para obter os dados da tabela, enviar os dados e registrar os resultados.

```
{
  PassData : function(result) {
    for(i=0;i<result.length;i++)
    {
      var firstRow = result[i];
      alert(firstRow.var_0);
    }
  }
}
```

A ilustração a seguir mostra o formulário como será exibido no momento da criação.



### ca\_pam\_getTableRowCount(\_id)

Retorna o número total de linhas de dados de uma tabela.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### Valor de retorno

Retorna um número de linhas como um número inteiro.

#### Exemplo

```
ca_pam_getTableRowCount('Form.employeeList')
```

Esse exemplo retorna a contagem de linhas do elemento de formulário Tabela da lista de funcionários.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## **ca\_pam\_setTableData(\_id, values)**

Define os dados de uma tabela.

### **Parâmetros de entrada**

#### ***\_id* (string)**

Especifica a identificação exclusiva de um elemento de formulário Tabela.

#### ***values* (object)**

Especifica os dados a serem exibidos nas linhas e nas colunas da tabela. Este parâmetro aceita somente o resultado da consulta de SQL.

### **Valor de retorno**

Nenhum.

### **Exemplo**

```
ca_pam_setTableData('Form.employeeList', result);
```

Este exemplo define a tabela employeeList para os valores que a variável de resultado retorna.

## ca\_pam\_setTableDataFromJSObject(*\_id*, *values*)

Preenche uma tabela com a matriz de objeto JavaScript.

### Parâmetros de entrada

#### *\_id* (string)

Especifica a ID da tabela.

#### *values* (object)

Especifica a matriz do JavaScript que contém uma lista de objetos. As propriedades dos objetos devem ser as mesmas que o nome da coluna para preencher os dados.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo mostra como preencher uma tabela a partir de uma matriz de objeto de JavaScript. A ID da tabela é Form.name e tem duas colunas, firstName e lastName.

```
var tableData = new Array();
var data1 = new Object();
data1.firstName = "firstName1";
data1.lastName = "lastName1";
tableData[0] = data1;
```

```
var data2 = new Object();
data2.firstName = "firstName2";
data2.lastName = "lastName2";
tableData[1] = data2;
ca_pam_setTableDataFromJSObject('Form.names', tableData );
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## Referência de objeto

### ca\_pam\_getTextFieldValue(\_id)

Retorna um valor a partir de um campo Texto.

#### Parâmetros de entrada

##### ***\_id*** (string)

Especifica a identificação exclusiva de um campo Texto.

#### Valor de retorno

O *valor* do campo Texto (string).

#### Exemplo

Este exemplo retorna o valor do campo Texto de top\_vendors.

```
ca_pam_getTextFieldValue('Form.top_vendors')
```

Este exemplo retorna o valor do campo Texto de top\_vendors e define o campo Texto de dynamic\_field como o valor retornado.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue para as pastas de Formulários de interação do usuário e, em seguida:
  - 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.
  - 08 Preencher tabela SOAP WS: Preencher tabela SOAP WS.

## ca\_pam\_setTextFieldValue(\_id, val)

Define o valor de um campo Texto.

### Parâmetros de entrada

#### ***\_id*** (string)

Especifica a identificação exclusiva do campo para o qual definir um valor específico.

#### ***val*** (string)

Define o novo valor para o campo Texto.

### Valor de retorno

Nenhum.

### Exemplo

Este exemplo define o campo de texto top\_vendors como "Eu voto no fornecedor 3 porque:"

```
ca_pam_setTextFieldValue('Form.top_vendors', 'I vote for vendor 3 because: ')
```

Para continuar este exemplo, use ca\_pam\_getTextFieldValue('Form.top\_vendors') depois que os usuários inserirem seus motivos para enviar suas declarações completas.

```
ca_pam_setTextFieldValue('Form.dynamic_field',  
ca_pam_getTextFieldValue('Form.top_vendors'))
```

Este exemplo retorna o valor do campo top\_vendors e define o dynamic\_field como este valor.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Origem dos dados

### ca\_pam\_getDatasetData(dataSetExpression,callBack)

Recupera o valor de uma variável a partir de um conjunto de dados do CA Process Automation.

**Observação:** nas releases do CA Process Automation antes de 04.1.00, este método era chamado ca\_pam\_getDataFromDatasetQuery.

**Parâmetros de entrada*****datasetExpression (string)***

Define uma sequência de caracteres, expressão ou chamada de função que resulta em um caminho absoluto para um conjunto de dados. Este parâmetro inclui a palavra-chave Conjuntos de dados, o caminho da biblioteca para um objeto do conjunto de dados e o nome da variável de matriz ValueMap.

```
Datasets [ "/BR412-DATA-FOLDER-20111214/Folder_FORMS_20120328/dataset_forms_demo/My_Dataset55" ].MyVal
```

***callback (object)***

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos onSuccess e onFailure.

**Valor de retorno**

Quando a função recupera os dados, a função onSuccess(result) ou onFailure(caught) callback é executada. A própria função não retorna qualquer valor.

**Exemplo**

```
ca_pam_getDatasetData(ca_pam_getTextFieldValue('Form.TxtFieldWithDatasetPath'), callback)
```

O código de exemplo contém as seguintes informações:

- O campo de texto TxtFieldWithDatasetPath armazena a expressão do conjunto de dados.
- A função getTextFieldValue obtém o valor da expressão de conjunto de dados.
- A função getDataUsingDatasetExpression obtém os valores do conjunto de dados e chama a função callback para determinar o que acontecerá em seguida.
- Para continuar com este exemplo, é possível usar as seguintes funções na declaração callback.onSuccess para aplicar os dados em uma tabela ou em um campo de pesquisa, respectivamente:

```
ca_pam_setTableData('Form.employeeList', result);
ca_pam_showDataInTable(result, 'Form.Results', 'Dataset results');
```

**ca\_pam\_getSQLData(driverName,connectionURL,userName,password,query,callback)**

Direciona o SQL para recuperar dados de uma origem de dados externa e armazenar esses dados em uma variável (denominada callback) definida por você.

**Observação:** nas releases do CA Process Automation antes de 04.1.00, este método era chamado ca\_pam\_getDataFromSQLQuery.

### Parâmetros de entrada

#### ***driverName*** (string)

Define o nome da classe do driver SQL instalado que você está usando (por exemplo, com.mysql.jdbc.driver).

#### ***connectionURL*** (string)

Define o URL do aplicativo de banco de dados a ser consultado, conforme mostrado no exemplo a seguir:

```
jdbc:mysql://myPC-xp.myCompany.com:<porta>/<caminho> ou  
https://<servidor>:<porta>/
```

#### ***userName*** (string)

Define um nome de usuário ou credenciais de logon para uma conta de usuário predefinida com permissões suficientes para executar a consulta.

#### ***password*** (string)

Define a senha associada ao nome de usuário especificado.

#### ***query*** (string)

Define uma instrução de consulta SQL específica ou uma referência a uma instrução de consulta SQL. O código a seguir representa um exemplo de consulta SQL:

```
"selecione * no employeedatatable, em que empName como  
'%"+ca_pam_getTextFieldValue('Form.empName')+"%";
```

#### ***callback*** (object)

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos onSuccess e onFailure.

### Valor de retorno

Quando a função recupera os dados, a função onSuccess(result) ou onFailure(caught) callback é executada. A própria função não retorna qualquer valor.

### Exemplo

```
ca_pam_getSQLData(ca_fd.js.driverName  
( ), ca_fd.js.connectionURL( ), ca_fd.js.userName( ), ca_fd.js.passwo  
rd( ), ca_fd.js.queryEmployeeName( ), callback);
```

Este exemplo consulta uma origem de dados externa e armazena os dados resultantes na variável callback.

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário e, em seguida, 06 Preencher BD do menu suspenso: Preencher BD do menu suspenso.

## Análise de JSON

### ca\_pam\_convertJSONToJSObject(jsonString)

Use esse método para criar um objeto JavaScript a partir de uma sequência de caracteres JSON. O método `ca_pam_convertJSONToJSObject` oferece suporte à notação de aspas. A notação de aspas é uma forma alternativa de acessar qualquer propriedade a partir de um objeto JavaScript quando o nome da propriedade não é um identificador válido de JavaScript. Por exemplo, para acessar a propriedade `id` do objeto `book`, a sintaxe da notação de aspas é `book["id"]`.

#### Parâmetros de entrada

##### *jsonString* (string)

Define a sequência de caracteres JSON para converter em um objeto JavaScript.

#### Valor de retorno

Retorna o objeto JavaScript.

#### Exemplo

O exemplo mostra uma sequência de caracteres JSON:

```
{ "menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      { "value": "New", "onclick": "CreateNewDoc()" },
      { "value": "Open", "onclick": "OpenDoc()" },
      { "value": "Close", "onclick": "CloseDoc()" }
    ]
  }
}
```

O script de exemplo a seguir acessa o valor do campo de identificação:

```
var parsedJSON = ca_pam_convertJSONToJSObject(jsonString);
var menuObj = parsedJSON.menu.;
var idValue = menuObj.id.;
```

O script de exemplo a seguir acessa o elemento `onclick` no segundo elemento da matriz `menuitem`:

```
var parsedJSON = ca_pam_convertJSONToJSObject(jsonString);
var menuObj = parsedJSON.menu;
var popUpObj = menuObj .popup;
var menuItemArray = popUpObj .menuitem.;
var secondElementOfMenuItemArray = menuItemArray[1];
var onclickElement = secondElementOfMenuItemArray.onclick;
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.

## Métodos REST

### `ca_pam_getRESTData(url, doNotValidateCert, headers, callBack)`

Usa o método HTTP get para iniciar o serviço REST. Você pode usar esse método com as seguintes assinaturas:

```
ca_pam_getRESTData(url, callBack);
```

```
ca_pam_getRESTData(url, doNotValidateCert, callBack);
```

#### Parâmetros de entrada

##### ***url*** (string)

Define o URL da solicitação HTTP. O URL começa com `http://` ou `https://`.

##### ***doNotValidateCert*** (boolean)

Especifica se um certificado SSL válido é encontrado. Esse campo é relevante ao consultar um URL HTTPS.

- `false` - valida o certificado SSL e faz com que a operação falhe se o certificado for inválido.
- `true` - aceita o certificado SSL, mesmo que ele seja inválido, e continua a fazer a chamada HTTP.

##### ***headers*** (object)

Define uma lista de pares de chave/valor que define os cabeçalhos na solicitação.

##### ***callBack*** (object)

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos `onSuccess` e `onFailure`.

#### Valor de retorno

Quando a função recupera os dados, a função `onSuccess(result)` ou `onFailure(caught)` `callBack` é executada. A própria função não retorna qualquer valor.

#### Exemplo

É possível localizar essa função no conteúdo pronto no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.

## **ca\_pam\_getRESTDataAuth(url, userName, password, doNotValidateCert, headers, callBack)**

Inicia o serviço REST em um URL especificado que exige autenticação. Esse método oferece suporte apenas à autenticação HTTP básica. Você pode usar esse método com as seguintes assinaturas:

```
ca_pam_getRESTDataAuth (url, userName, password, callBack)
```

```
ca_pam_getRESTDataAuth (url, userName, password, doNotValidateCert, callBack)
```

### **Parâmetros de entrada**

#### **url (string)**

Define o URL da solicitação HTTP. O URL começa com http:// ou https://.

#### **username (string)**

Define o nome de usuário para o qual autenticar o URL especificado.

#### **password (string)**

Especifica a senha associada ao nome de usuário especificado.

#### **doNotValidateCert (boolean)**

Especifica se um certificado SSL válido é encontrado. Esse campo é relevante ao consultar um URL HTTPS.

- false - valida o certificado SSL e faz com que a operação falhe se o certificado for inválido.
- true - aceita o certificado SSL, mesmo que ele seja inválido, e continua a fazer a chamada HTTP.

#### **headers (object)**

Define uma lista de pares de chave/valor que define os cabeçalhos na solicitação.

#### **callBack (object)**

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos onSuccess e onFailure.

### **Valor de retorno**

Quando a função recupera os dados, a função onSuccess(result) ou onFailure(caught) callBack é executada. A própria função não retorna qualquer valor.

## **ca\_pam\_postRESTData(url, body, contentType, doNotValidateCert, headers, callBack)**

Usa o método HTTP post para iniciar o serviço REST. Você pode usar esse método com as seguintes assinaturas:

```
ca_pam_postRESTData (url, callBack)
ca_pam_postRESTData (url, body, callBack)
ca_pam_postRESTData (url, body, contentType, callBack)
ca_pam_postRESTData (url, body, contentType, doNotValidateCert,
callBack)
```

### **Parâmetros de entrada**

#### **url (string)**

Define o URL da solicitação HTTP. O URL começa com http:// ou https://.

#### **body (string)**

Define os dados a serem enviados na solicitação HTTP.

#### **contentType (string)**

Define o tipo de conteúdo que compõe o corpo da solicitação HTTP. Esse valor é enviado como um cabeçalho (tipo de conteúdo) na solicitação HTTP.

#### **doNotValidateCert (boolean)**

Especifica se um certificado SSL válido é encontrado. Esse campo é relevante ao consultar um URL HTTPS.

- false - valida o certificado SSL e faz com que a operação falhe se o certificado for inválido.
- true - aceita o certificado SSL, mesmo que ele seja inválido, e continua a fazer a chamada HTTP.

#### **headers (object)**

Define uma lista de pares de chave/valor que define os cabeçalhos na solicitação.

#### **callBack (object)**

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos onSuccess e onFailure.

### **Valor de retorno**

Quando a função recupera os dados, a função onSuccess(result) ou onFailure(caught) callBack é executada. A própria função não retorna qualquer valor.

## **ca\_pam\_postRESTDataAuth(url, userName, password, body, contentType, doNotValidateCert, headers, callback)**

Inicia o serviço REST em um URL especificado que exige autenticação. O servidor chama os métodos REST de maneira assíncrona.

A função `onSuccess(result)` ou `onFailure(caught)` do objeto `callback` é executada conforme apropriado após os dados serem recuperados do servidor. O resultado pode ser qualquer formato de dados retornado pelo serviço REST. O método oferece suporte apenas à autenticação HTTP básica. Você pode usar esse método com as seguintes assinaturas:

```
ca_pam_postRESTDataAuth(url, userName, password, callback)
ca_pam_postRESTDataAuth(url, userName, password, body, callback)
ca_pam_postRESTDataAuth(url, userName, password, body, contentType,
callback)
ca_pam_postRESTDataAuth(url, userName, password, body, contentType,
doNotValidateCert, callback)
```

### **Parâmetros de entrada**

#### ***url* (string)**

Define o URL da solicitação HTTP. O URL começa com `http://` ou `https://`.

#### ***username* (string)**

Define o nome de usuário para o qual autenticar o URL especificado.

#### ***password* (string)**

Define a senha associada ao nome de usuário especificado.

#### ***body* (string)**

Define os dados a serem enviados na solicitação HTTP.

#### ***contentType* (string)**

Define o tipo de conteúdo que compõe o corpo da solicitação HTTP. Esse valor é enviado como um cabeçalho (tipo de conteúdo) na solicitação HTTP.

#### ***doNotValidateCert* (boolean)**

Especifica se um certificado SSL válido é encontrado. Esse campo é relevante ao consultar um URL HTTPS.

- `false` - valida o certificado SSL e faz com que a operação falhe se o certificado for inválido.
- `true` - aceita o certificado SSL, mesmo que ele seja inválido, e continua a fazer a chamada HTTP.

#### ***headers* (object)**

Define uma lista de pares de chave/valor que define os cabeçalhos na solicitação.

#### ***callback* (object)**

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos onSuccess e onFailure.

### Valor de retorno

Quando a função recupera os dados, a função onSuccess(result) ou onFailure(caught) callBack é executada. A própria função não retorna qualquer valor.

### Exemplo

É possível localizar essa função no conteúdo pronto no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 07 Preencher tabela RESTful WS: Preencher tabela RESTful WS.

## Exemplo: Método REST POST

Para recuperar todos os formulários de solicitação inicial da biblioteca do CA Process Automation, chame a API PAM REST contida no CA Catalyst Container.

1. Crie um formulário de solicitação de interação e o elabore como mostra a ilustração a seguir:

Name	Value
Name	Var_1
Description	
Disabled	
Hidden	
Hide Label	
Label	
Maximum Length	
Minimum Length	
Password	
Pattern	
Pattern Message	
Required	
Style	
Tab Index	
Text Direction	
_id	Form.Var_1
onBlur	ca_fd.js.populateDataInTable(

2. Adicione o campo Palavra-chave ao formulário de solicitação de interação.
3. Adicione o texto a seguir na seção Script do formulário de solicitação de interação:

```
{
  sample : function()
  {
    var callBack = new Object();
    callBack.onSuccess = function(result)
    {
      alert(result);
    }

    callBack.onFailure = function(caught)
    {
      alert(caught);
    }

    var headers = new Object();
    var contentType="application/xml";
    ca_pam_getDataFromRESTPostHTTPAuthentication('https://<hostName>:<PortNumber>
/node/rest/CA:00074:01/_ops/QueryStartRequests',useName
password,<requestBody>,contentType,true,headers,callBack);
  }
}
```

O corpo da solicitação é o seguinte:

```
<QueryStartRequestFormsRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="p1:QueryStartRequestFormsRequest"
      xmlns:p1="http://ns.ca.com/2011/09/pam-ops" >
  <Filter>
    <LookUpPath>/</LookUpPath>
    <IsRecursive>true</IsRecursive>
  </Filter>
</QueryStartRequestFormsRequest>
```

4. Forneça o valor `ca_fd.js.sample()` ao evento `onBlur` do campo Palavra-chave.
5. Clique em Salvar.

O script é executado de acordo com o evento `onBlur` do campo Palavra-chave. O formulário de solicitação inicial é recuperado de acordo com o corpo da consulta especificado e a solicitação REST realizada.

## Métodos SOAP

### **ca\_pam\_getSOAPData(serviceURL, methodName, inlineText, soapVersion, stripXMLNamespacesFromResponse, callBack)**

Faz uma chamada de serviço web e converte o XML resultante em um objeto JavaScript.

Trata-se de um método sobrecarregado. `ca_pam_getSOAPData` pode ser chamado sem fornecer a versão SOAP e/ou o parâmetro `stripXMLNamespacesFromResponse`. Se você não fornecer esses parâmetros, a versão SOAP será considerada `SOAP_1_1` e `stripXMLNamespacesFromResponse` será considerada `true`.

A seguir estão as assinaturas de método sobrecarregado:

- `ca_pam_getSOAPData (serviceURL,methodName, inlineText,callBack)`
- `ca_pam_getSOAPData (serviceURL,methodName, inlineText,soapVersion, callBack)`

#### **Parâmetros de entrada**

##### ***serviceURL (string)***

Especifica o URL para o serviço SOAP. O URL é geralmente acessado por HTTP ou HTTPS. O URL é um ponto de entrada para um ou mais métodos.

##### ***methodName (string)***

Especifica o método ou a função a ser executado. A função passa o método para o serviço SOAP como um cabeçalho MIME SOAPAction.

##### ***inlineText (string)***

Seleciona a origem da solicitação de entrada do serviço SOAP. Este parâmetro inclui uma mensagem XML, a qual pode incluir um envelope SOAP.

##### ***soapVersion (string)***

Especifica a versão do servidor SOAP em que a chamada é feita. Os valores possíveis incluem `SOAP_1_1` ou `SOAP_1_2`. O padrão é `SOAP_1_1`.

##### ***stripXMLNamespacesFromResponse (boolean)***

Especifica se os Espaços para nomes da resposta devem ser removidos da resposta XML (verdadeiro). O padrão é verdadeiro.

##### ***callBack (object)***

Define uma referência para o objeto de função a ser chamada após a recuperação dos dados, incluindo seus métodos `onSuccess` e `onFailure`.

A função `ca_pam_getSOAPData` chama o método `callBack` de maneira assíncrona no servidor. Quando a função recupera os dados, a função `onSuccess(result)` ou `onFailure(caught) callBack` é executada. Se a função retornar um erro de SOAP, o método `onFault(faultString)` será chamado e o resultado será uma sequência de caracteres XML.

Se você não fornecer a versão SOAP e os parâmetros `stripXMLNameSpacesFromResponse`, seus valores padrão serão substituídos e a resposta será retornada ao objeto `callback` no `onSuccess(result)`.

**Valor de retorno**

Quando a função recupera os dados, a função `onSuccess(result)` ou `onFailure(caught) callback` é executada. Se a função retornar um erro de SOAP, o método `onFault(faultString)` será chamado e o resultado será uma sequência de caracteres XML. A própria função não retorna qualquer valor.

**Observação:** é possível localizar essa função no conteúdo pronto no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

*Exemplo: método SOAP*

Este exemplo usa o método de serviço web `getStartRequestForms` do CA Process Automation para preencher uma tabela com todos os Formulários de solicitação inicial na Biblioteca. O método de serviço web `getStartRequestFormsweb` retorna todos os formulários de solicitação inicial em uma pasta especificada.

**Observação:** esse método usa o método `ca_pam_convertXMLToJSObject(xmlString, elementTagName)` (descrito mais adiante nesta seção) para criar um objeto JavaScript a partir do XML.

1. Crie um formulário de solicitação de interação e o elabore como mostra a ilustração a seguir:

Nome	Valor
Nome	Var_0
Desativado	
Descrição	
Oculto	
Rótulo da caixa de sel...	Check Box
Style	
Tab Index	
_id	Form.Var_0
onBlur	
onClick	ca_fd.js.populateDateInTable()
onFocus	
onMouseDown	
onMouseMove	
onMouseOut	
onMouseOver	
onMouseUp	

2. Adicione os campos a seguir ao formulário de solicitação de interação:

**soapService**

Defina esse campo com o seguinte URL:

http://hostname:portNumber/itpam/soap

**soapAction**

Defina esse campo como getStartRequestForms.

**soapData**

Preencha essa área de texto com o seguinte código:

```
<tns:getStartRequestForms xmlns:tns="http://www.ca.com/itpam">
  <tns:auth>
    <!--xsd:Choice Type-->
    <tns:token>token__</tns:token>
    <tns:user>pamadmin</tns:user>
    <tns:password>pamadmin</tns:password>
  </tns:auth>
  <tns:filter>
    <tns:lookUpPath isRecursive="true"></tns:lookUpPath>
  </tns:filter>
</tns:getStartRequestForms>
```

**retrieveSRF**

Defina o atributo onClick desta caixa de seleção com o valor ca\_fd.js.retrieveSRFAndPopulateTable().

3. Crie uma tabela denominada srfs.
4. Adicione nome e campos de texto refPath como colunas à tabela srfs.
5. Adicione o seguinte texto na seção Script do formulário:

```
{
  retrieveSRFAndPopulateTable: function()
  {
    var callBack = new Object();
    callBack.onSuccess = function(result)
    {
      var srfResult = ca_pam_ convertXMLToJSObject
      (result, 'startRequest');
      var tableArray = new Array();
      for( i=0;i<srfResult .length;i++)
      {
        var object = new Object();
        object.name = srfResult [i]["name"];
        object.refPath = srfResult [i]["refPath"];
        tableArray[i] = object ;
      }
    }
  }
}
```

```
ca_pam_clearTableData('Form.srfs',0,ca_pam_getTableRowCount('Form.srfs')-1);
    ca_pam_setTableDataFromJSObject('Form.srfs',tableArray);
}
callback.onFailure = function(caught)
{
    alert(caught);
}

ca_pam_getSOAPData(ca_pam_getTextFieldValue('Form.soapService'),ca_pam_getText
TextFieldValue('Form.soapAction'),ca_pam_getTextFieldValue('Form.soapData'),'SOA
P_1_1',true,callback);
}
}
```

6. Clique em Salvar.
7. Para que o CA Process Automation verifique se a tabela srfs é preenchida com dados dinamicamente, marque a caixa de seleção.

O script é executado quando o valor da caixa de seleção é alterado. O script preenche a tabela srfs de forma dinâmica com os formulários de solicitação inicial retornados pela chamada SOAP. O método getSOAPData é usado para criar uma consulta SOAP para recuperar os dados, e o convertXMLToJSObject é usado para converter o XML de resposta em um objeto JavaScript. O objeto JavaScript pode então ser usado para preencher a tabela de forma dinâmica, usando o método ca\_pam\_setTableDataFromJSObject.

## Análise de XML

### ca\_pam\_convertXMLToJSObject(xmlString, elementTagName)

Cria um objeto JavaScript a partir do XML.

**Observação:** se o elemento XML tiver reconhecimento de espaço para nome e tiver um prefixo de espaço para nome, use a notação de aspas para acessá-lo.

#### Parâmetros de entrada

##### *xmlString* (string)

Define a sequência de caracteres XML para converter em um objeto JavaScript.

##### *elementTagName* (string)

(Opcional) Retorna o elemento que tem o nome da tag especificada. Se o elementTagName não for fornecido, o método retornará o elemento raiz.

#### Valor de retorno

Retorna um valor de tipo Objeto.

#### Exemplo

Considere o seguinte XML a ser convertido em um objeto JavaScript:

```
<?xml version="1.0"?>
<catalog>
  <book id="bk101">
    <author>Gambardella, Matthew</author>
    <title>XML Developer's Guide</title>
    <genre>Computer</genre>
    <price>44,95</price>
    <publish_date>2000-10-01</publish_date>
    <description>An in-depth look at creating applications
with XML.</description>
  </book>
</catalog>
```

Esse exemplo converte os dados XML acima em um objeto JavaScript.

- A sintaxe a seguir converte o documento XML em um objeto JavaScript:

```
var parsedXML = ca_pam_convertXMLToJSObject(xmlString);
```

- É possível usar qualquer uma das linhas a seguir para acessar o elemento de livro:

```
var bookObj = parsedXML . book[0];
var bookObj = parsedXML ["book"][0];
```

- É possível usar qualquer uma das linhas a seguir para recuperar o valor do atributo de identificação do elemento de livro:

```
var idVal = bookObj.id;
var idVal = bookObj["id"];
```

- É possível usar qualquer uma das linhas a seguir para acessar o valor de descrição do elemento de livro:

```
var bookDescription = bookObj.description[0]._text;  
var bookDescription = bookObj["description"][0]["_text"];
```

**Observação:** se um elemento tiver um atributo e um elemento com o mesmo nome, acesse-o como uma matriz. Na matriz, o primeiro integrante (índice 0) é sempre o atributo e os elementos a seguir.

### Exemplo

Considere o seguinte XML a ser convertido em um objeto JavaScript:

```
<?xml version="1.0"?>  
<categories>  
  <category name="Weather">  
    <name part="1">Sunny</name>  
    <name part="2">Rainy</name>  
  </category>  
  <category name="Climate">  
    <name part="1">Wet</name>  
    <name part="2">Dry</name>  
  </category>  
</categories>
```

Esse exemplo converte os dados XML acima em um objeto JavaScript.

- Use a sintaxe a seguir para recuperar o objeto da primeira categoria das categorias XML representadas como a xmlString:

```
var parsedXML= ca_pam_convertXMLToJSObject(xmlString);  
var firstCategory= parsedXML.category[0];
```

- Use a sintaxe a seguir para acessar o atributo de nome:

```
var bookName = firstCategory.name[0];
```

- Use a sintaxe a seguir para acessar o valor do elemento de primeiro nome:

```
var firstBookName = firstCategory.name[1]._text;
```

### Exemplo

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <Header/>
  <Body>
    <getStartRequestFormsResponse
xmlns="http://www.ca.com/itpam">
      <startRequests>
        <startRequest name="RahulSrf"
refPath="/Folder/">
          <description/>
        </startRequest>
        <startRequest name="Start Request Form"
refPath="/RestSupport/">
          <description/>
        </startRequest>
      </startRequests>
    </getStartRequestFormsResponse>
  </Body>
</Envelope>
```

Esse exemplo mostra o uso do método `ca_pam_convertXMLToJSObject(xmlString, elementTagName)` com o parâmetro `elementTagName`.

O script a seguir recupera todos os elementos `<startRequest>` do XML:

```
var srfResult =
ca_pam_convertXMLToJavaScriptObject(xmlString, 'startRequest');
srfResult é uma matriz de todos os elementos startRequest.
```

**Exemplo**

```
<?xml version="1.0"?>
<note _text="abc">
  <_text>Tove</_text>
</note>
```

Esse exemplo mostra um cenário no qual o XML apresenta `_text` como um elemento ou atributo. Nesse caso, o método cria uma matriz com o nome `"_text"`.

- O primeiro elemento da matriz retorna o nó de texto no elemento.
- O segundo elemento da matriz retorna o nó de atributo (com o nome `"_text"`) do elemento.
- Outros elementos da matriz retornam os nós do elemento filho (com o nome `"_text"`) do elemento.

O script a seguir acessa os valores:

```
var parsedXML= ca_pam_convertXMLToJavaScriptObject(xmlString);
var textNode= parsedXML["_text"][0];
var attributeValue= parsedXML["_text"][1];
var childTextNodeValue= parsedXML["_text"][2]["_text"];
```

O script contém o resultado a seguir:

```
textNode="";
```

(...como não há valor escalar com o nó).

```
attributeValue="abc";
childTextNodeValue="Tove"
```

Também é possível localizar a função no conteúdo predefinido no CA Process Automation.

1. Na página inicial, clique em Procurar conteúdo pronto.
2. Navegue até a pasta Formulários de interação do usuário, 08 Preencher tabela SOAP WS: Preencher tabela de SOAP WS.

## Criar um formulário simples com funções básicas

Neste exemplo, você está desenvolvendo um formulário para uma empresa que opera em três regiões principais (Califórnia, Pensilvânia e Carolina do Norte). Em cada Estado, a empresa tem um local de escritório principal e secundário identificado pelo nome da cidade. As partes interessadas responsáveis pela implementação da diretiva da empresa desejam que o formulário atenda aos seguintes requisitos de negócios:

- Todos os negócios originados na Califórnia devem ser roteados por padrão para o escritório de Anaheim.
- Todos os negócios originados na Pensilvânia devem ser roteados por padrão para o escritório da Filadélfia.
- Todos os negócios originados na Carolina do Norte Carolina devem ser roteados por padrão para o escritório de Raleigh.
- Todos os negócios originados em qualquer outro Estado devem ser roteados por padrão para o escritório de Pittsburgh.
- O usuário tem a opção de especificar outro Estado ou cidade.

### Exemplo: um formulário básico

#### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Crie um objeto Formulário de solicitação de interação ou localize um existente a ser usado para este exemplo de procedimento.
3. Clique duas vezes no objeto do formulário.  
A caixa de diálogo Formulário de solicitação de interação, ou Criador de formulário, é exibida.
4. Se o objeto do formulário ainda não estiver reservado, clique em Reservar.
5. No painel Elementos do formulário, expanda as seguintes entradas:
  - a. Elementos do formulário
  - b. Seu formulário (por exemplo, Interaction\_Request\_Form\_4)
  - c. Layout da página
  - d. Página

6. Clique em Página.
7. Clique em Renomear na barra de ferramentas
8. Insira o nome *Location*: e clique em OK.
9. Arraste um campo Seleccionar do painel Formulário para a página Local.
10. Repita a Etapa 9.  
Dois campos de seleção são exibidos no layout da página Location.
11. Realize as seguintes ações no painel Formulário:
  - a. Arraste três Opções de seleção para o primeiro campo Seleccionar.
  - b. Arraste seis Opções de seleção para o segundo campo Seleccionar.
12. Clique em cada um dos seguintes elementos do formulário e, em seguida, clique em Renomear para renomear cada objeto, conforme indicado:
  - a. Renomeie o primeiro campo de seleção para *Estado*.
    - Renomeie a primeira opção para *CA*.
    - Renomeie a segunda opção para *PA*.
    - Renomeie a terceira opção para *NC*.
  - b. Renomeie o segundo campo de seleção para *Cidade*.
    - Renomeie a primeira opção para *Anaheim*.
    - Renomeie a segunda opção para *Los Angeles*.
    - Renomeie a terceira opção para *Pittsburgh*.
    - Renomeie a quarta opção para *Filadélfia*.
    - Renomeie a quinta opção para *Charlotte*.
    - Renomeie a sexta opção para *Raleigh*.

13. Clique em cada um dos seguintes elementos do formulário e faça as seguintes configurações de propriedade no painel Propriedades (Nome e Valor).
  - a. Para Estado:
    - Defina Nome para *Estado*.
    - Defina o valor do evento onChange para `ca_fd.js.selectRegion()`.
    - Defina o valor da opção CA para *California*.
    - Defina o valor da opção PA para *Pennsylvania*.
    - Defina o valor da opção NC para *North Carolina*.
  - b. Para Cidade:
    - Defina Nome para *Cidade*.
    - Defina a propriedade Valor de cada opção de cidade para o nome da cidade. Por exemplo, defina Anaheim para *Anaheim*, Los Angeles para *Los Angeles* e assim por diante.
14. No painel Elementos do formulário, clique no nome do formulário e, em seguida, clique em Script, na barra de ferramentas.

A caixa de diálogo Script é exibida.

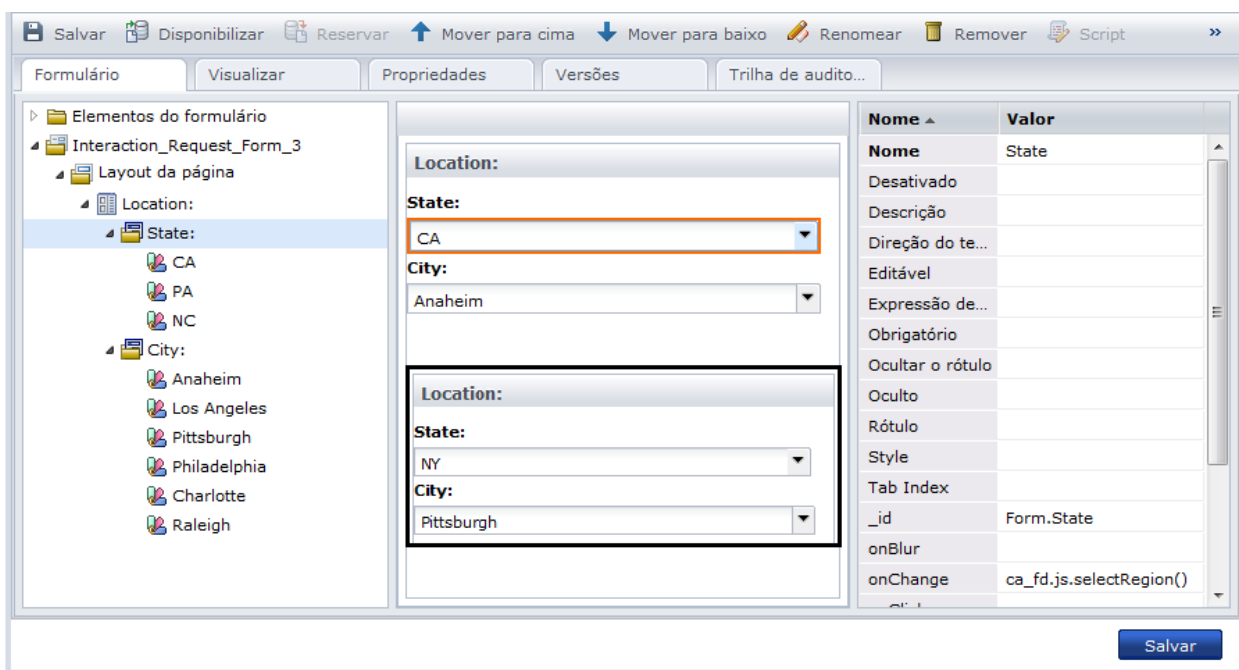
15. Copie e cole o seguinte código a seguir no [Editor de script](#) (na página 146):

```
{
selectRegion: function()
{
    var selectedState = ca_pam_getSelectedOptionValues('Form.State')[0];

    if(selectedState == 'North Carolina')
    {
        ca_pam_selectOption('Form.City', 'Raleigh', 'Raleigh', true);
    }
    else if(selectedState == 'California')
    {
        ca_pam_selectOption('Form.City', 'Anaheim', 'Anaheim', true);
    }
    else if(selectedState == 'Pennsylvania')
    {
        ca_pam_selectOption('Form.City', 'Philadelphia', 'Philadelphia', true);
    }
    else
    {
        ca_pam_selectOption('Form.City', 'Pittsburgh', 'Pittsburgh', true);
    }
}
}
```

16. Clique em Salvar. A caixa de diálogo Script é fechada.
17. Clique em Salvar no Criador de formulários.

18. Clique na guia Visualizar.
19. Confirme se os resultados são semelhantes aos resultados e aos valores de amostra na seguinte ilustração e tabela:



Estado:	Cidade padrão:
CA	Anaheim
PA	Philadelphia
NC	Raleigh
Qualquer outro Estado. Por exemplo, NY.	Pittsburgh

## Inicializar variáveis do formulário

Depois de criar um formulário de solicitação de interação, você pode defini-lo para um operador Atribuir tarefa de usuário específico em um processo. Você também pode adicionar o código para inicializar os campos do formulário em tempo de execução.

### Siga estas etapas:

1. Clique na guia Criador.
2. No criador de processos, abra ou crie um processo.
3. No grupo Controle de processo da paleta Operadores, arraste um operador Atribuir tarefa de usuário para o seu processo.
4. Clique duas vezes no operador Atribuir tarefa de usuário para abrir a paleta Propriedades.
5. Na paleta Propriedades:
  - a. Expanda Tarefa de usuário.
  - b. No campo Formulário de solicitação de interação, clique no botão de pesquisa para procurar um formulário. Clique em OK.
  - c. Clique no campo Código de inicialização de dados do formulário para expandi-lo.
6. Na janela Código de inicialização de dados do formulário, inicialize qualquer variável do formulário. Para os exemplos a seguir, `myTextField` é o `_id` do elemento do formulário que você deseja inicializar.
  - Para tipos de dados simples, digite:

```
Form.myTextField='welcome'.
```
  - Se o campo simples estiver em um `ValueMap` ou conjunto de campos, digite:

```
Form.value_map= newValueMap();  
Form.valuemap.myTextField="welcome";
```
  - Se o campo simples estiver em um mapa de valores complexo em um `valueMap`, digite:

```
Form.value_map.value_map_nested= newValueMap();  
Form.value_map.value_map_nested.text_field_nested="test";
```
7. Na janela Código de inicialização de dados do formulário, clique em OK.



# Capítulo 8: Recursos

---

Um objeto de *recurso* é um modelo que representa elementos da arquitetura de seu sistema. Use-o para sincronizar processos independentes que dependem de elementos comuns da infraestrutura e para quantificar e controlar o acesso a entidades de TI específicas. Inclua vários recursos que representem entidades relacionadas em um único objeto Recurso.

Use os objetos Recurso para:

- Balancear a carga de processamento em todos os processos em execução em um touchpoint.
- Sincronizar a execução de processos que não podem ser executados em paralelo.
- Implementar os bloqueios no nível de ambiente que ativam ou desativam simultaneamente vários recursos.
- Gerenciar processos e sistemas estrategicamente com direitos de segurança comuns.

Você agrupa os recursos porque eles estão relacionados entre si de alguma maneira. Bancos de dados compartilhados, links de transmissão, acesso simultâneo a um número limitado de licenças de software, processos simultâneos em um touchpoint, cotas numéricas e outros recursos são alguns exemplos. Depois de avaliar o desempenho, você poderá alocar os recursos do sistema para processos necessários às tarefas de missão crítica. Você pode limitar o número de conexões de FTP simultâneas usadas pelo CA Process Automation. Você poderá usar recursos para iniciar um processo sucessor *somente* depois que um processo antecedente liberar um recurso. Os recursos também podem ser usados para representar e controlar o acesso a uma entidade específica do ambiente de TI, como um arquivo de log que recebe atualizações de vários processos.

Esta seção contém os seguintes tópicos:

[Como os recursos funcionam](#) (na página 370)

[Criar um objeto Recurso](#) (na página 371)

[Edite um objeto Recurso](#). (na página 372)

[Monitorar e editar recursos](#) (na página 374)

[Adicionar um operador Gerenciar recursos em um processo](#) (na página 375)

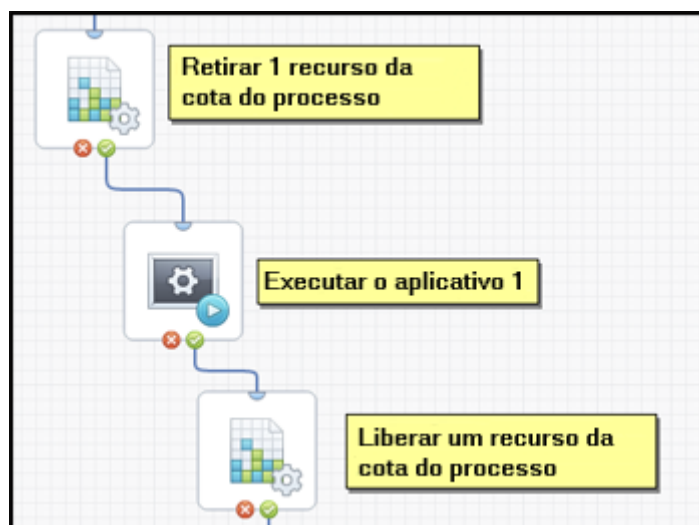
[Definir ações de recursos](#) (na página 376)

[Verificar e responder a recursos indisponíveis](#) (na página 377)

## Como os recursos funcionam

O operador *Gerenciar recursos* está no grupo do operador Controle de processo. Um processo pode usar o operador Gerenciar recursos para assumir as unidades disponíveis de um recurso especificado. Se todas as unidades do recurso especificado forem assumidas (no mesmo processo ou em processos diferentes), o operador Gerenciar recursos atrasará o processamento nessa ramificação até que o recurso tenha unidades livres.

A figura a seguir mostra um processo assumindo uma unidade do recurso *cota de processo* antes de executar um aplicativo. Se houver unidades disponíveis no recurso cota de processo, o processamento continuará no operador *Executar o aplicativo 1*. Se não houver nenhuma unidade disponível, o processamento não é iniciado até que uma unidade se torne disponível ou o tempo limite do operador seja atingido. Depois que o operador Executar aplicativo 1 for concluído, a unidade do recurso será liberada de volta para o recurso cota de processo e o processamento continuará com os operadores subsequentes no processo.



Os recursos permitem definir o número de unidades disponíveis e quantas unidades são consumidas. Um recurso consiste em um número máximo de unidades, o valor atual das unidades disponíveis e um sinalizador que indica se o recurso está bloqueado.

É possível executar as ações a seguir com os recursos:

- Assumir uma quantidade especificada de unidades de um recurso disponível. Inserir um valor maior no campo Usado.
- Liberar uma quantidade especificada de unidades de um recurso disponível. Inserir um valor menor no campo Usado.
- Bloquear um recurso
- Desbloquear um recurso

Um operador Gerenciar recursos pode consumir ou liberar qualquer número de recursos especificado. Como desenvolvedor ou administrador, você pode usar os operadores Gerenciar recursos para ajustar o balanceamento de carga em um touchpoint específico. Ele também pode bloquear um recurso para evitar o consumo das unidades de recurso por qualquer outro processo.

As três restrições a seguir aplicam-se aos recursos:

- O número máximo de unidades de qualquer recurso é um valor arbitrário que é possível calibrar e ajustar para os seus requisitos do sistema. O número de recursos que um processo utiliza também é arbitrário. O limite máximo é 9.999; no entanto, oriente-se pelo desempenho e pela arquitetura. Aloque unidades de recurso para os processos que melhor atendam aos seus requisitos de implementação.
- O valor das unidades de recurso usado no momento é sempre menor ou igual ao valor máximo do recurso.
- Um processo dependente de recurso deve aguardar até que seu número especificado de unidades esteja disponível.
- Os operadores não podem consumir unidades de um recurso bloqueado.
- Os operadores não podem bloquear um recurso bloqueado por outro processo.

## Criar um objeto Recurso

Crie e defina quantos objetos *Recurso* forem necessários em cada orquestrador. O operador Recurso executa operações de recurso em um processo. O operador Recurso usa as versões atuais dos objetos Recurso especificados. Qualquer modificação feita no objeto Recurso em seu estado atual estará imediatamente disponível ao operador Recurso na disponibilização.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. No painel esquerdo, clique em uma pasta, selecione Novo e, em seguida, escolha Recursos.

Um novo recurso é exibido.

3. Clique no nome do recurso e renomeie-o para um nome mais significativo.
4. Edite o objeto Recurso.

**Observação:** você também pode criar objetos Recurso dinamicamente usando código.

## Edite um objeto Recurso.

Edite um objeto Recurso para gerenciar os recursos individuais dentro dele. Você também pode gerenciar versões, exibir propriedades e examinar o histórico do objeto.

Adicione entradas de recursos individuais para aplicativos, conexões ou outras ocorrências que deseje controlar. Isso lhe permitirá:

- Definir o número máximo de instâncias que pode ser executadas em qualquer momento.
- Rastrear o número de instâncias ou unidades executadas ao mesmo tempo.
- Rastrear o número de instâncias disponíveis ou de unidades liberadas que podem ser iniciadas a qualquer momento.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique duas vezes em um objeto Recurso.  
A caixa de diálogo Recursos é exibida. A guia Recursos é aberta por padrão.
3. Clique em Adicionar para adicionar um recurso individual.
4. Examine qualquer valor nas colunas da guia Recursos. Clique nas células editáveis para inserir novos valores.

#### Nome

Lista os nomes dos recursos individuais em um objeto Recurso.

#### Quantidade

Lista o número total de unidades atribuídas a um recurso. Uma unidade é um número arbitrário que funciona como uma cota em um processo.

#### Usado

Indica o número de unidades atribuídas.

#### Livre

Indica o número de unidades não atribuídas. Definido pela fórmula:

Livre = Quantidade - Usado

#### Estado

Especifica se um recurso está bloqueado ou desbloqueado. Clique no ícone de cadeado nesta coluna para alterar o estado bloqueado ou desbloqueado de um recurso. Você também pode usar um operador Recurso para bloquear um recurso de forma programática em uma programação ou em um processo. Outros operadores Gerenciar recurso não poderão bloquear, desbloquear, assumir ou retornar unidades de recurso para um recurso bloqueado até que o bloqueio seja liberado. Um bloqueio permite que um processo ou uma programação monopolize um recurso enquanto processa os operadores.

#### **% de uso**

Posicione o mouse sobre o indicador visual para exibir a porcentagem numérica do recurso que está sendo utilizado. Qualquer parte restante está liberada.

#### **Descrição**

Uma breve descrição do recurso. A coluna descrição permite que digitar o texto que descreve um recurso.

5. Clique em Disponibilizar ou em Salvar e Fechar.

#### **Observações:**

- No campo *Quantidade*, especifique a quantidade do recurso. As cotas para os operadores nos processos são obtidas a partir desse número. A quantidade é um valor arbitrário que não está relacionados sozinho às unidades de qualquer computador real ou recurso do sistema (como CPU, memória ou largura de banda). É possível usá-la para dividir um recurso entre os processos em qualquer maneira necessária. Não há regras sobre a quantidade de um recurso. É possível especificar a quantidade *1*, de modo que somente uma instância de um operador que usa muitos recursos de CPU possa ser executada por qualquer processo em qualquer momento específico.
- As variáveis de conjunto de dados podem ser usadas para definir o uso de recursos, para que o uso possa ser ajustado em um touchpoint sem que seja necessário abrir e configurar os processos que utilizam um recurso. Por exemplo, se você definir a quantidade para *100*, poderá alterar uma variável de uso de *10* para *20*, *50* ou até mesmo para *100* a fim de acomodar as demandas em um touchpoint.
- As colunas *Usado* e *Livre* mostram quantas unidades de um recurso estão sendo usadas ou estão disponíveis no momento. No campo *Usado*, você pode inserir um valor entre 0 e o número total de unidades mostradas no campo *Quantidade*. Mais comumente, um operador Recurso muda essas configurações de forma programática em um processo ou em uma programação.

## Monitorar e editar recursos

É possível monitorar e editar os recursos em um objeto Recurso na página Operações. A página Operações permite que você modifique a versão atual de um objeto Recurso. Suas alterações são aplicadas automaticamente a todos os operadores Gerenciar recursos usando o objeto Recurso.

### Siga estas etapas:

1. Clique na guia Operações.
2. No painel Links:
  - a. Expanda Exibição de processos ou Recursos.
  - b. Expanda a lista de pastas ou pesquise uma exibição de processos ou objeto Recurso específico.
  - c. Selecione um objeto Recurso.

A versão atual do objeto Recurso é exibida.
3. Selecione um recurso e, em seguida, faça qualquer uma das alterações a seguir.
  - a. Na barra de ferramentas, clique em Bloquear para bloquear o recurso. Clique em Desbloquear para desbloquear um recurso. Você também pode clicar no ícone de cadeado na coluna Estado para alternar esta configuração.
  - b. Clique em Limpar para liberar os recursos usados e redefinir os recursos livres para a quantidade máxima disponível.
  - c. Clique nos campos Quantidade, Usado e Descrição para editar seus valores.
  - d. Clique em Atualizar para exibir os últimos dados do sistema.

Suas alterações estão disponíveis para os operadores Gerenciar recursos.
4. Para adicionar ou editar recursos em um operador Gerenciar recursos em uma programação:
  - a. Expanda Links e clique em Programações ativas ou Programações globais.
  - b. Na lista de programações, clique duas vezes em uma programação.
  - c. No item Gerenciar recursos da programação, clique em Propriedades.
  - d. Clique na guia Específico.
  - e. Clique em Reservar.
  - f. Clique em Adicionar, Editar ou Excluir para configurar os recursos.
  - g. Clique em Disponibilizar ou clique em Salvar e Fechar.

## Adicionar um operador Gerenciar recursos em um processo

Depois de criar um objeto de recurso contendo um ou mais recursos, adicione um operador Recurso a um processo. Por exemplo, coloque um operador Recurso antes e outro após outros operadores para equilibrar a carga. O primeiro operador Recurso utiliza ou assume os recursos e o segundo operador libera ou retorna as unidades, disponibilizando-as para outros processos em espera.

### Siga estas etapas:

1. Abra um processo no Criador de processos.
2. Abra a paleta do operador Controle de processo de seu recurso.
3. Arraste e solte o operador Gerenciar recursos em um local no processo.
4. Defina os links de entrada e saída. O operador Gerenciar recursos possui quatro possíveis links de saída:
  - **Concluído** é processado quando o operador é bem-sucedido. A variável *Resultado* é definida como *1* e a variável *Motivo* é definida como *CONCLUÍDO*.
  - O link **Falha** é processado quando o Módulo do intérprete não é capaz de concluir o operador com êxito. A variável *Resultado* é definida como *-1* e a variável *Motivo* é definida como *FALHA*.
  - O link **Tempo limite** será processado se o operador Recurso não for concluído dentro de um intervalo de tempo limite opcional especificado. A variável *Resultado* é definida como *1* e a variável *Motivo* é definida como *TEMPO LIMITE*.
  - O link **Personalizar resultado** é processado quando as configurações de execução determinam o resultado. A variável *Resultado* é definida como *0* e a variável *Motivo* é definida como *PERSONALIZADO*.
5. Clique duas vezes no operador Recurso para configurar as opções.  
As paletas Conjunto de dados e Propriedades do recurso são exibidas.

## Definir ações de recursos

Você pode definir a ação que deseja que cada operador Gerenciar recursos execute. As ações possíveis incluem assumir e liberar unidades ou bloquear e desbloquear recursos.

### Siga estas etapas:

1. No Criador de processos, clique duas vezes em um operador Gerenciar recursos.
2. Na paleta Propriedades, clique no botão **Adicionar**. É possível criar várias ações para diversos recursos dentro de um operador Recurso.

A caixa de diálogo Propriedades da ação é exibida.

3. No campo do objeto Recursos, especifique o objeto de recurso que você deseja usar. É possível inserir o caminho completo para o objeto na biblioteca de automação ou clicar no botão Procurar para localizar o objeto.
4. (Opcional) Clique em Abrir para exibir ou editar os recursos no objeto.
5. No campo de nome do recurso, insira o nome do recurso ou uma expressão.

**Observação:** os campos Caminho do recurso e Nome do recurso aceitam expressões. Insira qualquer sequência de caracteres literal entre aspas duplas.

6. No campo Ação, selecione a ação que deseja que o recurso execute na lista suspensa:

#### **Obter unidades**

Assume o número de unidades de recurso especificado no campo *Quantidade*.

#### **Liberar unidades**

Disponibiliza o número de recursos especificados no campo *Quantidade*.

#### **Bloquear recurso**

Bloqueia o recurso para que outros operadores de recursos não possam assumir as unidades de recurso ou bloquear o recurso. As ações ainda podem liberar as unidades de recurso que foram assumidas antes de um recurso ser bloqueado, mas as unidades livres não estarão disponíveis até que o recurso seja desbloqueado.

#### **Desbloquear recurso**

Desbloqueia um recurso bloqueado.

7. Ao assumir ou liberar unidades de recurso, especifique uma quantidade no campo Quantidade.
8. Clique em Salvar e fechar.

A nova ação é adicionada à lista Ação.

**Observações:** para remover uma ação da lista Ação, clique na ação e, em seguida, clique no botão **Excluir**. Para editar uma ação existente, clique na ação e, em seguida, clique no botão **Editar**. Para exibir uma ação selecionada em um processo que não está reservado, clique no botão **Exibir**.

O gráfico a seguir mostra dois exemplos.



## Verificar e responder a recursos indisponíveis

As propriedades a seguir determinam quais recursos são executados pelo operador e como ele reage quando os recursos estão indisponíveis:

- Campo *Tempo limite*
- Caixa de seleção *Todos os recursos devem estar disponíveis*
- Caixa de seleção *Executar ações*

## Especifique um intervalo de tempo limite

Às vezes, uma ação tenta assumir mais unidades de um recurso do que estão disponíveis ou tenta bloquear um recurso já bloqueado. Nessas situações, defina o intervalo de *Tempo limite* para determinar o tempo que o operador aguardará pela disponibilidade dos recursos. Depois que o intervalo de tempo limite expirar, o operador verificará se o recurso está disponível em um horário final antes de expirar. Depois que um operador Recurso expirar, o módulo de processos processará o link de saída de tempo limite no operador.

### Siga estas etapas:

1. Abra um processo.
2. Arraste um operador Gerenciar recursos para o processo.
3. Clique duas vezes no operador Gerenciar recursos para exibir suas propriedades.
4. Na paleta Configurações de execução, configure as opções disponíveis no grupo Tempo limite.

### Sem tempo limite

Quando essa opção está marcada, o operador aguarda indefinidamente até que todos os recursos, incluindo todas as ações listadas em *Ações*, possam ser executados e estejam disponíveis.

Quando essa opção está desmarcada, a configuração Duração ou Data de destino referente ao tempo limite é aplicada.

### Tipo

Selecione Duração ou Data de destino. O operador aguarda até atingir o tempo limite. Por exemplo, para especificar um intervalo de tempo limite, selecione *Duração* e digite o número de segundos. Quando você especifica uma duração de tempo limite de *0 segundos*, o operador não espera. Se os recursos estiverem indisponíveis, ele expirará imediatamente. O operador somente terá êxito se os recursos estiverem imediatamente disponíveis.

### Duração/data e hora de destino

Especifique a quantidade numérica de segundos ou uma data e hora fixa para ser usada como o período de tempo limite.

### Ação

Selecione Continuar, Redefinir, Cancelar ou Abandonar. Se você escolher Cancelar, o operador processará o link de saída *Falha*.

5. Salve o processo.

## Especificar as configurações de Ação e de Disponibilidade de recursos

As propriedades a seguir determinam os recursos que são executados pelo operador. Elas também determinam como o operador responde quando os recursos não estão disponíveis. Defina a caixa de seleção *Todos os recursos devem estar disponíveis* para determinar como o CA Process Automation lidará com a disponibilidade de recursos. Defina a opção *Executar ações* para determinar como o CA Process Automation funcionará com relação à disponibilidade de recursos e às ações predefinidas.

### Siga estas etapas:

1. Abra um processo.
2. Arraste um operador Gerenciar recursos para o processo.
3. Clique duas vezes no operador Gerenciar recursos para exibir suas propriedades.
4. Defina a caixa de seleção *Todos os recursos devem estar disponíveis*:

#### Marcado

Especifica que todos os recursos que as ações listaram no campo *Ação* devem estar disponíveis antes da aplicação de qualquer ação. O operador somente será bem-sucedido se todos os recursos forem disponibilizados nas restrições da configuração de *Tempo limite*.

#### Limpar

Permite que o operador conclua somente as ações para as quais os recursos estão disponíveis. O operador será bem-sucedido se uma ou mais ações listadas forem executadas com êxito nas restrições da configuração de *Tempo limite*.

5. Defina a caixa de seleção *Executar ações*:

**Marcado**

O operador executa todas as ações, se possível.

**Limpar**

O operador não executa nenhuma ação. Se os recursos estiverem disponíveis dentro das restrições de *Tempo limite*, o operador executará o link de saída *Com êxito* sem executar nenhuma ação.

6. Salve o processo.

**Observação:** essas configurações podem ser usadas com um recurso definido para ativar ou desativar um conjunto inteiro de processos. Antes de iniciar suas tarefas, esses processos verificam se não há nenhum bloqueio no recurso tentando obter uma única unidade de recurso por meio do recurso. Dependendo do resultado do teste, outro mecanismo pode bloquear ou desbloquear o recurso, como:

- Tarefas de programação (em que a ativação ou a desativação dos processos se baseiam em restrições de horário)
- Tarefas iniciadas manualmente (usando um Formulário de solicitação inicial)
- Um processo iniciado por um aplicativo externo de monitoramento
- Um processo que monitora uma condição interna ou externa em um loop

## Verificar a disponibilidade do recurso sem executar ações

Defina a opção *Executar ações* para determinar como o CA Process Automation funcionará com relação à disponibilidade de recursos e às ações predefinidas.

# Capítulo 9: Calendários, programações, tarefas e disparadores

---

Em geral, qualquer processo pode ser programado por uma tarefa *Executar processo* em um objeto *Programação* de acordo com as datas válidas definidas em um objeto *Calendário*. Uma *tarefa* em uma programação especifica um operador selecionado para execução em um determinado touchpoint. Os *disparadores* permitem que os aplicativos externos iniciem um processo.

Esta seção descreve os calendários, as programações, as tarefas e os disparadores.

Esta seção contém os seguintes tópicos:

[Calendários](#) (na página 381)

[Programações](#) (na página 397)

[Gerenciamento de tarefas](#) (na página 407)

[Administrar disparadores](#) (na página 410)

## Calendários

Os calendários definem as regras para datas que são aplicadas a tarefas, de modo que elas sejam executadas ou não quando esperado. Por exemplo, você pode criar um objeto Calendário chamado *LastOpenDayofMonth* e usá-lo para programar backups completos e relatórios mensais no último dia disponível de cada mês.

Depois de definir os objetos *Calendário*, use-os em objetos *Programação* para determinar quando as tarefas devem ser executadas. A programação de tarefas ou processos em determinados dias exige um objeto *Programação*. As programações coordenam horários de tarefas ou processos com as datas válidas definidas por um calendário.

Você pode criar vários calendários e associar qualquer calendário único a qualquer número de tarefas programadas. A separação de calendários das programações permite definir regras comuns para datas que podem ser reutilizadas em vários contextos de programação. Para alterar as regras (por exemplo, aquelas que representam os dias fechados) de todas as tarefas que usam (incluem ou excluem) um calendário, edite o calendário que descreve essas datas. As programações aplicam automaticamente qualquer alteração feita em um calendário.

## Criar um objeto Calendário

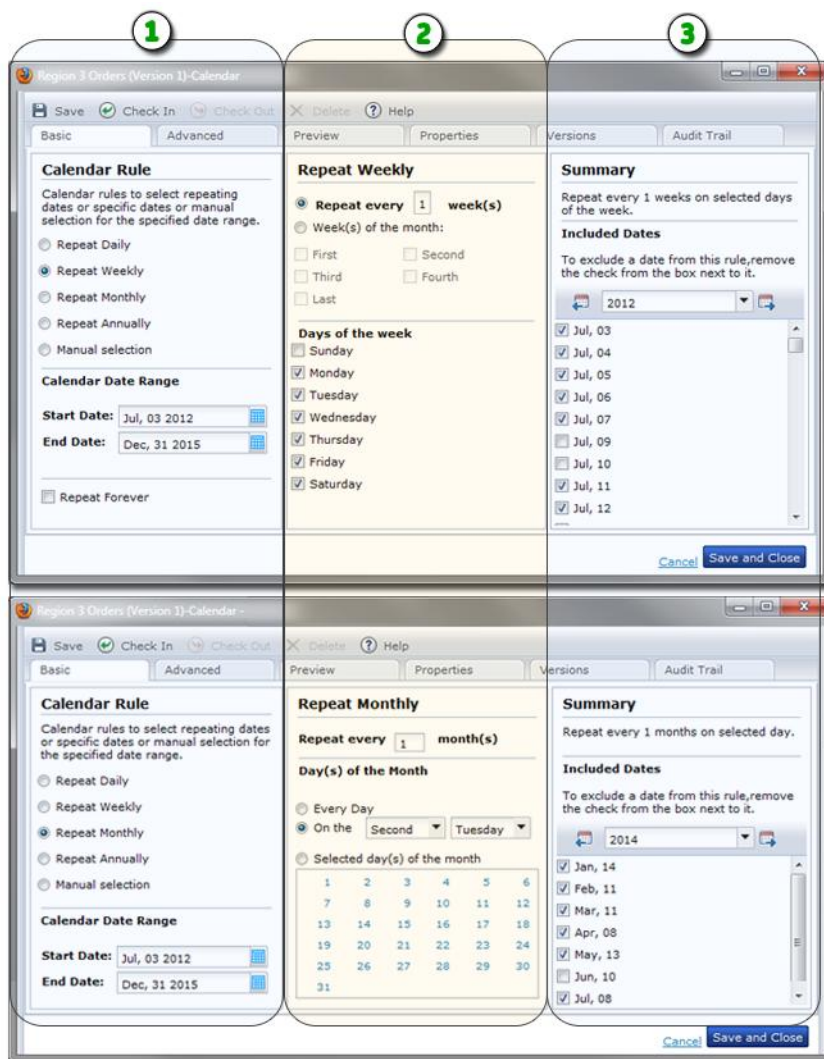
É possível criar um objeto Calendário em qualquer pasta no Navegador da biblioteca.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique em uma pasta.
3. Na barra de ferramentas, clique em Novo, selecione Objeto e escolha Calendário.  
Um novo objeto Calendário é exibido.
4. Clique no nome do calendário para renomeá-lo,
5. Clique duas vezes no calendário para editá-lo.  
O Criador de calendários é aberto.
6. Defina regras de calendário que o aplicativo usa para criar um calendário de datas incluídas e excluídas. Escolha um dos seguintes itens:
  - Clique na guia Básico para definir as regras básicas.
  - Clique na guia Avançado para definir as regras avançadas.

## Criador de calendários básico

O Criador de calendários é aberto quando você edita um objeto Calendário. Use a guia Básico para definir regras que geram datas incluídas. Em seguida, você pode excluir determinadas datas. Outra abordagem é criar um calendário especificamente para as datas excluídas que você deseja aplicar e gerenciar separadamente.




---

### Item: Descrição:



**Recorrência de regra de calendário:** selecione uma opção que representa o padrão de recorrência para as datas do calendário que você deseja definir. As opções são Diariamente, Semanalmente, Mensalmente, Anualmente ou suas próprias datas selecionadas manualmente. Você também pode definir um intervalo de datas ou definir a regra para repetir indefinidamente.

Item:	Descrição:
2	<b>Configurações detalhadas:</b> essa parte da guia Básico baseia sua aparência na sua seleção de Regra do calendário. Por exemplo, uma recorrência semanal resulta em configurações detalhadas para os dias da semana e semanas do mês. Como outro exemplo, uma recorrência mensal mostra as configurações para determinados dias do mês.
3	<b>Resumo de datas incluídas e excluídas:</b> essa área lista todos os dias na sua regra de calendário. Desmarque a caixa de seleção de uma data para excluí-la do calendário.

## O Criador de calendários avançado

As regras de calendário avançadas são estruturas de árvore hierárquica. A guia Avançado para um objeto Calendário sempre mostra três conjuntos de regras de calendário:

- Datas incluídas manualmente
- Datas excluídas manualmente
- Regra do Calendário

Esses três elementos raiz para uma regra de calendário são fixos. Esses três conjuntos não podem ser excluídos. Defina regras criando condições de data nos elementos raiz. Uma regra de calendário combina operadores de data (como dia do mês, mês do ano ou semana do ano) com operadores lógicos (interseção, união, exclusão e como). A regra do calendário é, na verdade, o operador de união raiz para todas as regras do calendário.

Inicie a criação de uma regra de calendário arrastando uma regra organizada pelo ícone em Todas as regras para o conjunto Regra do calendário. Também é possível incluir ou excluir datas manualmente.

Depois de colocar uma regra em Regras de calendário, configure suas propriedades.

As propriedades para um objeto de condição de data na regra do calendário também têm propriedades que você pode expandir para configurar a condição de data, como as propriedades de intervalo de meses.

Por exemplo, se as propriedades do intervalo de meses define a etapa para “3”, especifica que a cada três meses em um ano é válido. Porque o intervalo começa em janeiro e termina em dezembro, esse intervalo de três meses repete durante o ano. Essa regra define uma condição para executar tarefas a cada trimestre.

O exemplo de tarefa a seguir mostra essa condição em uma regra e adiciona uma condição que especifica o quinto dia de cada mês. O operador de interseção funciona como um operador lógico E para combinar essas duas condições em uma única regra que especifica o quinto dia de cada trimestre.

**Exemplo: Criar uma regra que especifica o quinto dia de cada trimestre**

1. Arraste o operador de interseção para Regra do calendário.
2. Arraste o intervalo de meses em Interseção.
3. Clique no operador de intervalo de meses. Defina um intervalo de meses de janeiro a dezembro com a etapa definida como 3.
4. Arraste um operador de intervalo de dias e especifique um intervalo de 5 a 5 com a etapa definida como 1.
5. Para salvar as alterações para a versão de trabalho do objeto Calendário, clique em Salvar e fechar.
6. Para testar o calendário, clique na guia Visualizar.

É possível continuar a adicionar outros operadores e condições para definir uma regra. A hierarquia de lógica define a ordem em que as condições são aplicadas.

Em termos de uma equação booleana, é possível tirar foto de um objeto Calendário como parênteses delimitadores ao redor e aplicando restrições booleanas para operadores e condições subordinadas.

- O operador Union aplica um booliano OU para operadores e condições subordinadas.
- O operador interseção aplica um booliano E para operadores e condições subordinadas.
- O operador exclusão aplica um booliano NÃO a uma condição excluída.
- O operador Como insere outra regra de calendário em uma equação, para que você possa pensar como inserir uma função booleana criada pelo usuário. Uma regra do calendário ou combinação de regras de calendário podem lidar com praticamente qualquer problema de programação.

Observe que, dependendo das necessidades de um determinado problema de programação, existem várias maneiras para criar regras de calendário. Você também pode definir um conjunto de datas em um calendário separado. É possível usar o operador Como para especificar o objeto Calendário e adicioná-lo diretamente a Regra do calendário. Também é possível usar um operador de união, interseção ou exclusão para incluir um operador Como em uma regra de calendário.

## Operadores lógicos de regra do calendário

Use os quatro operadores de conjunto lógico para incluir e excluir datas em regras de calendário. Os operadores *União*, *Interseção* e *Exclusão* fornecem uma maneira para incluir e excluir datas representadas por condições de data básica ou por combinações de condições e operadores de ramificação. A lista a seguir descreve cada operador lógico:



### União

Indica que uma ou mais das condições vinculadas devem ser atendidas para que a condição combinada seja satisfeita.

Coloque uma ou mais ramificações ou condições básicas sob esse ícone.



### Interseção

Indica que todas as condições vinculadas devem ser atendidas para que a condição combinada seja satisfeita.

Coloque uma ou mais ramificações ou condições básicas sob esse ícone.



### Exclusão

Indica uma condição de básico ou uma ramificação a ser excluída de uma regra.

Todas as datas que não estão de outra forma selecionadas são excluídas. Por esse motivo, isso é útil somente para excluir os dias que eles estão selecionados por outra parte da regra. Por exemplo, nenhuma finalidade é servida excluindo as terças-feiras, a menos que elas sejam definidas como dias válidos por outras condições e operadores em uma regra. Portanto, se uma condição especifica a semana de trabalho (de segunda a sexta) como dias válidos, é possível usar o operador de exclusão para excluir terças-feiras deste conjunto.

Expanda o operador Exclusão para mostrar as ramificações Incluído e Excluído.

Clique em uma ramificação e, em seguida, adicione uma condição ou um operador para definir as datas incluídas ou excluídas. Esse operador tem dois conjuntos de argumentos ramificados:

**Incluído:** uma ou mais condições básicas ou ramificações que representam as datas a serem incluídas na regra.

**Excluído:** uma ou mais condições básicas ou ramificações que representam as datas a serem excluídas das datas definidas pelo conjunto Incluído.



### Como

Use o operador Como para utilizar um conjunto existente de datas definidas por outro objeto de calendário em sua regra. Esse operador tem os seguintes parâmetros:

**Nome do calendário:** o nome do calendário referenciado.

**Delta:** alterna as datas válidas definidas pelo calendário referenciado pelo número de dias especificado. Digite um número negativo para mover as datas para trás ou um número positivo para mover as datas para frente.

**Dias em aberto:** quando marcada, essa opção indica que o delta ou o turno se aplicam apenas a dias em aberto.

Por exemplo, uma nova regra do calendário pode fazer referência a outro calendário especificando os dias de reserva com um delta igual a 1. A condição resultante na nova regra do calendário especifica o dia imediatamente após os dias de reserva.

## Operadores de datas de regras do calendário

Esta seção descreve as condições elementares em datas e seus parâmetros. As condições podem ser colocadas no painel de regra do Criador de calendários. Para selecionar uma data nas propriedades de qualquer operador, clique no ícone de calendário para abrir o visualizador de calendário e selecionar uma data.



### Lista de datas

Especifica datas individuais. Por exemplo:

- 1º de março de 2014
- 15 de julho de 2014
- 23 de setembro de 2015

### Parâmetros

- Uma lista de datas com anos.
- Para adicionar uma data, clique no botão Adicionar.
- Para excluir uma data da lista, selecione a data e clique no botão Excluir.
- Clique nos botões Mover para cima e Mover para baixo para reordenar as datas na lista.



### Intervalo de datas

Especifica um intervalo diário, semanal ou mensal em um intervalo de datas, do início ao fim.

Por exemplo, todas as semanas a partir de 1º de março de 2015 até 1º de julho de 2016.

### Parâmetros

- **Início:** a data de início do intervalo.
- **Fim:** a data de término do intervalo.
- **Repetir para sempre:** marque essa caixa de seleção para ignorar a data de término e estender o intervalo indefinidamente.
- **Etapa:** indica a quantidade de unidades (dias, semanas ou meses) em cada intervalo. Por exemplo, um intervalo com uma unidade de semana e uma etapa padrão de 1 ocorre uma vez na semana 1, novamente na semana 2 e uma terceira vez na semana 3. Quando Etapa for definida como 3, o intervalo ocorrerá uma vez nas semanas de 1 a 3, novamente nas semanas de 4 a 6 e uma terceira vez nas semanas de 7 a 9.
- **Unidade:** especifica a frequência de recorrência ou o intervalo. Selecione Dia, Semana ou Mês.



### Data sem lista de anos

Especifica uma lista de datas de aniversário explícitas.

Essa condição é normalmente usada para especificar os feriados que caem no mesmo dia de cada ano. Por exemplo, 1º de janeiro e 25 de dezembro.

#### Parâmetros

- Uma lista de datas sem anos.
- Para adicionar uma data, clique no botão Adicionar parâmetro (+). Clique no botão procurar (...) no novo parâmetro para adicionar o controle de calendário Seleccionar data e selecione uma data.
- Para excluir uma data da lista, selecione a data e, em seguida, clique no botão Excluir parâmetro (x). Você pode clicar nos botões Mover para Cima e Mover para Baixo para reordenar as datas na lista.



#### Data sem intervalo de ano

Especifica um intervalo de datas de aniversário sem o ano.

Por exemplo, de 21 de março a 20 de junho (para primavera).

#### Parâmetros

- **Início:** a data de início sem um ano para o intervalo.
- **Fim:** a data de término sem um ano para o intervalo.



#### Intervalo anual

Especifica um intervalo de anos.

Você pode especificar os anos bissextos, iniciando um intervalo em um ano bissexto e especificando uma etapa 4 (por exemplo, 2000 a 2024 com uma etapa 4).

#### Parâmetros

- **Início:** o ano de início do intervalo.
- **Fim:** o ano de término do intervalo.
- **Etapas:** o número de anos a partir de um ano válido até o próximo ano válido.



#### Intervalo de meses

Especifica um ou mais meses do ano.

O primeiro semestre é especificado com um intervalo de 1 a 6 com uma etapa de 1. O segundo semestre é especificado com um intervalo de 7 a 12 com uma etapa de 1.

#### Parâmetros

- **Início:** o mês de início do intervalo.
- **Fim:** o mês de término do intervalo.
- **Etapas:** o número de meses a partir de um mês válido até o próximo mês válido.



#### Semana do intervalo mensal

Especifica um ou mais semanas do mês.

O CA Process Automation implementa as normas ISO para semanas parciais. Uma semana que cruza com um determinado mês é considerada parte do mês se quinta-feira da semana cai no mês.

Por exemplo, se 1º de junho é uma sexta-feira, a primeira semana do mês inicia em 4 junho. Se 1º de junho for uma quarta-feira, a primeira semana do mês inicia em 30 de maio.

É possível ter a "primeira segunda-feira do mês" e não ser "segunda-feira a primeira semana do mês". Para definir a antiga, é mais simples para combinar as condições "dia do mês" e "dia da semana".

#### Parâmetros

**Início:** a semana de início do intervalo.

**Fim:** a semana de término do intervalo.

**Etapa:** o número de semanas a partir de uma semana válida até a próxima semana válida.

**Reversão:** a contagem começa com a última semana do mês e retrocede.



#### Semana do intervalo anual

Especifica uma ou mais semanas do ano.

O CA Process Automation implementa as normas ISO para semanas parciais. Uma semana que cruza com um determinado ano é considerada parte do ano se a quinta-feira da semana cai no ano.

Por exemplo, se 1º de janeiro é uma sexta-feira, a primeira semana do ano começa em 4 de janeiro. Se 1º de janeiro for uma quarta-feira, a primeira semana do ano começa em 30 de dezembro do ano anterior.

Portanto, é possível que a "primeira segunda-feira do ano" não seja "segunda-feira da primeira semana do ano". Para definir a antiga, é mais simples combinar as condições "dia do ano" e "dia da semana".

#### Parâmetros

**Início:** a semana de início do intervalo.

**Fim:** a semana de término do intervalo.

**Etapa:** o número de semanas a partir de uma semana válida até a próxima semana válida.

**Reversão:** a contagem começa com a última semana do ano e retrocede.



#### Intervalo de dia

Especifica um intervalo de dias válidos (de 1 a 31) em um mês com um dia inicial, um dia final e uma etapa.

Também é possível especificar que a iteração inicie do final do mês ou que apenas os dias em aberto sejam contados em cada etapa. Os Dias em aberto são os que não foram especificados por uma condição ou regra que fecha ou exclui as datas.

Por exemplo, o último dia do mês é especificado pelo intervalo começando e terminando com 1 com reverter selecionado. O último dia da semana do mês deve ser especificado quando a caixa de seleção aberta também está marcada e um intervalo de dias da semana especificando de segunda a sexta-feira for adicionado com um operador e.

#### Parâmetros

**Início:** o dia de início do intervalo.

**Fim:** o dia de término do intervalo.

**Etapas:** o número de dias a partir de um dia válido até o próximo dia válido.

**Reversão:** a contagem começa com o último dia do mês e retrocede.

**Dias em aberto:** a contagem em etapas inclui apenas os dias em aberto quando os dias estão fechados por uma condição ou uma regra.



#### Dia do intervalo de ano

Especifica um intervalo de dias válidos (de 1 a 366) em um ano com um dia inicial, um dia final e uma etapa. O dia 366 é válido em anos bissextos.

Também é possível especificar que a iteração inicie no final do ano ou que apenas os dias em aberto sejam contados em cada etapa. Os Dias em aberto são os que não foram especificados por uma condição ou regra que fecha ou exclui as datas.

Por exemplo, é possível especificar inverno como o intervalo de 21 de dezembro a 20 de março.

Ou com exemplo um pouco mais complicado, para especificar a cada 10 dias em todo o ano, é possível usar um intervalo de 1 a 365 (ou 366 para um salto) com uma etapa de 1. É possível especificar os últimos dez dias em aberto do ano com um dia inicial de 1, um dia final de 10, com reverter e aberto selecionado.

#### Parâmetros

**Início:** o dia de início do intervalo.

**Fim:** o dia de término do intervalo.

**Etapas:** o número de dias a partir de um dia válido até o próximo dia válido.

**Reversão:** a contagem em etapas começa com o último dia do ano e retrocede.

**Em aberto:** a contagem em etapas inclui apenas os dias em aberto.



#### Dia do intervalo de semana

Especifica um ou mais dias da semana (de segunda a domingo) como um intervalo, com um dia inicial, um dia final e uma etapa.

Por exemplo, os fins de semana são especificados pelo intervalo começando no sábado e terminando em domingo com uma etapa de 1.

### Parâmetros

**Início:** o dia de início do intervalo.

**Fim:** o dia de término do intervalo.

**Etapa:** o número de dias a partir de um dia válido até o próximo dia válido.



### Dia da semana do mês

Especifica um dia da semana em uma semana indexada de um determinado mês. A semana é indexada no início ou no final do mês.

### Parâmetros

**Dia da semana:** especifica o dia da semana.

**Mês:** especifica o mês para o qual o dia da semana é aplicável.

**Índice da semana:** especifica o índice da semana para o qual o dia da semana é aplicável. (O valor pode ser de 1 a 5, pois não existe um mês com mais de 5 semanas)

**Reversão:** se você marcar essa caixa de seleção, a contagem do índice da semana será iniciada pela última semana.

Por exemplo, é possível selecionar segunda-feira como um dia da semana, setembro como um mês e 3 como um índice da semana: em setembro, a terceira segunda-feira é incluída no calendário. Se marcou a caixa de seleção inversa, em setembro, a terceira segunda-feira do último está incluída no calendário.



### Dia da semana do ano

Especifica um dia da semana em uma semana indexada do ano. A semana é indexada no início ou no final do ano.

### Parâmetros

**Dia da semana:** especifica o dia da semana.

**Índice da semana:** especifica o índice da semana para o qual o dia da semana é aplicável. (O valor pode ser de 1 a 53 porque em um ano não ter mais do que 53 semanas)

**Reversão:** se você marcar essa caixa de seleção, a contagem do índice da semana será iniciada pela última semana.

Por exemplo, é possível selecionar segunda-feira como um dia da semana, 43 como o índice da semana, quarenta e três segundas-feiras do ano são incluídas no calendário. Se marcou a caixa de seleção inversa, as quarenta e três segundas-feiras na última semana são incluídas no calendário.

## Adicionar e remover as datas do calendário manualmente

Você pode, às vezes, necessitar de datas em um objeto Calendário que não são facilmente especificadas por uma regra de calendário. De forma semelhante, uma regra pode incluir datas que, por algum motivo, você não deseja em um calendário. Você pode usar Datas incluídas manualmente e Datas excluídas manualmente do painel Regras de calendário selecionadas para adicionar ou remover as datas selecionadas manualmente.

### Para adicionar ou remover datas

1. Abra um calendário.
2. Clique na guia Avançado.
3. Expanda Datas incluídas manualmente ou Datas excluídas manualmente.
4. No visualizador de mês no painel Propriedades, clique com o botão direito do mouse em uma data selecionada e clique em um dos comandos para incluir ou excluir datas no menu de atalho.
  - Para incluir datas específicas para um ano específico, clique no comando incluir data completa.
  - Para incluir datas de aniversário para todos os anos, clique no comando incluir data de aniversário.
  - Para excluir datas específicas para um ano específico, clique no comando excluir data completa.
  - Para excluir datas de aniversário para todos os anos, clique no comando excluir data de aniversário.

O comando Incluir data completa está disponível apenas quando as datas excluídas estão selecionadas no painel. O comando Excluir data completa está disponível apenas quando as datas incluídas estão selecionadas no painel.

## Criador de calendários: guia Visualizar

Use a guia Visualizar para inspecionar as datas a serem incluídas em um calendário e excluídas dele.

Também é possível comparar as datas em um calendário com as datas em um segundo calendário. Por exemplo, defina um calendário de trabalho padrão que omita os feriados ou os dias de férias que você definiu em outro *calendário de exclusão*. É possível visualizar como o calendário de exclusão afetaria o calendário de trabalho padrão à medida que você o editar.

calendario (Versão 2)-Calendário - Mozilla Firefox

Salvar Disponibilizar Reservar Ajuda

Básico Avançado Visualizar Propriedades Versões Trilha de audito...

2014

**Datas incluídas**

- 1 Jan, 2014
- 2 Jan, 2014
- 3 Jan, 2014
- 4 Jan, 2014
- 6 Jan, 2014
- 7 Jan, 2014
- 8 Jan, 2014
- 9 Jan, 2014
- 10 Jan, 2014
- 11 Jan, 2014
- 13 Jan, 2014
- 15 Jan, 2014
- 17 Jan, 2014
- 18 Jan, 2014
- 20 Jan, 2014
- 21 Jan, 2014
- 22 Jan, 2014
- 23 Jan, 2014
- 24 Jan, 2014
- 25 Jan, 2014
- 27 Jan, 2014
- 28 Jan, 2014
- 29 Jan, 2014
- 30 Jan, 2014
- 31 Jan, 2014
- 1 Fev, 2014
- 3 Fev, 2014
- 4 Fev, 2014
- 5 Fev, 2014
- 6 Fev, 2014
- 7 Fev, 2014
- 8 Fev, 2014
- 10 Fev, 2014
- 11 Fev, 2014
- 12 Fev, 2014

**janeiro 2014**

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

**fevereiro 2014**

D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	

**março 2014**

D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

**abril 2014**

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

**maio 2014**

D	S	T	Q	Q	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

**junho 2014**

D	S	T	Q	Q	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

**julho 2014**

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

**agosto 2014**

D	S	T	Q	Q	S	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

**setembro 2014**

D	S	T	Q	Q	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

**outubro 2014**

D	S	T	Q	Q	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

**novembro 2014**

D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

**dezembro 2014**

D	S	T	Q	Q	S	S
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Visualizar calendário de exclusões

/FOLDER-20111214/Region 3 Days Off

Delta 0  Dias em aberto

Máx turnos 1

Atualizar

Item:	Descrição:
1	<b>Visualizar guia e ano:</b> após definir um calendário na guia Básico ou Avançado, clique na guia Visualizar para exibir as datas. Clique em Ano anterior, Próximo ano ou selecione um ano.
2	<b>Datas incluídas:</b> esse painel exibe todas as datas incluídas nas definições de regras do seu calendário.
3	<b>Visualizar calendário de exclusão:</b> (opcional) selecione um calendário separado para exibir os conflitos em negrito vermelho no calendário de visualização.
4	<b>Datas incluídas:</b> a visualização do calendário exibe as datas incluídas no seu calendário com números em negrito azul escuro.
5	<b>Datas excluídas:</b> a visualização do calendário exibe as datas que são omitidas manual ou automaticamente das regras do calendário com números em azul claro.
6	<b>Conflito de datas:</b> a visualização do calendário exibe as datas que se sobrepõem ou estão em conflito com as datas definidas por um calendário de exclusão opcional com números em negrito vermelho.
7	<b>Campos de resolução de conflito:</b> use o campo Delta para especificar o número de dias que uma data elegível é modificada quando cai em uma data omitida ou excluída. Um valor Delta negativo alterna para frente (mais cedo) e um valor positivo alterna para trás (mais tarde). Quando esse valor for zero (o padrão), a data elegível, normalmente incluída na regra do calendário, estará marcada em negrito vermelho e será omitida.  Marque a caixa de seleção Dias em aberto para contar apenas os dias incluídos ao mudar a programação de modo a evitar uma data excluída ou omitida. Os dias em aberto são os que não foram especificados por uma condição ou regra que omita ou exclua as datas. Se a caixa de seleção Dias em aberto não estiver marcada, uma data modificada terá a possibilidade de cair em outro dia excluído ou omitido.  Use o campo Máx turnos definir o número máximo de turnos ou ajustes que são permitidos se turnos repetidos caírem em dias fechados.

## Excluir calendários

Dias fechados são os dias em que um grupo de tarefas programadas não pode ser executado. Os dias fechados podem ser especificados em um calendário (por exemplo, os finais de semana são implicitamente fechados quando uma regra especifica dias da semana) ou em um calendário de férias separado. Um calendário de férias é criado com regras que especificam datas válidas, como qualquer outro calendário. A especificação de um calendário como um calendário de férias ou de exclusão fecha as datas que poderiam, de outra maneira, ser definidas como datas válidas para a execução de tarefas.

Por exemplo, algumas tarefas não podem ser executadas em feriados empresariais. Nesse caso, é possível criar um calendário que especifica todos os feriados empresariais. Em seguida, para cada tarefa em uma programação que você deseja ignorar em feriados empresariais, especifique o calendário de feriados empresariais como o *calendário de exclusão*. Os feriados empresariais são dias fechados para as tarefas.

## Programações

Os objetos de *Programação* são configurados quando as tarefas do processo ou do operador são executadas. Especifique dias válidos na programação ou por referência a objetos *Calendário* definidos anteriormente. Os objetos de Programação permitem agrupar, coordenar e programar quando as tarefas são executadas com relação aos elementos organizacionais ou arquiteturais de uma empresa. Por exemplo:

- Aplicativos
- Propriedade
- Monitoramento
- Manutenção
- Processos funcionais

Os objetos de programação especificam:

- As tarefas (processos e operadores) a serem executados
- A hora do dia em que cada tarefa é iniciada.
- O intervalo de repetição para várias ocorrências de uma tarefa
- Os dias em que as tarefas são iniciadas (usando um calendário, especificando dias explícitos ou uma combinação de ambos)
- Os dias em que as tarefas não têm permissão para iniciar (usando calendários de férias e/ou especificando dias excluídos)
- A hora do dia em que cada tarefa é encerrada.

É possível criar uma programação com ou sem especificar um calendário. Processos e outras tarefas executados diariamente ou em dias especificados não exigem um objeto de calendário. Para programar datas usando as regras de calendário:

1. Crie objetos de calendário.
2. Especifique os calendários nos objetos de programação.

Os calendários definem regras que especificam datas válidas para a execução de tarefas e dias fechados nos quais as tarefas não podem ser executadas. Os objetos de programação associam tarefas relacionadas operacionalmente a um calendário e especificam os horários em que as tarefas são executadas nos dias válidos definidos pelas regras do calendário.

*Equation 2: Este gráfico destaca as funções da página Programações.*

The screenshot shows a web-based interface for task scheduling. On the left, there is a sidebar with two main sections: 'Processos' (Processes) and 'Operadores' (Operators). The 'Processos' section is expanded to show a tree view under 'Process\_Module', with 'Test' selected. The 'Operadores' section is expanded to show 'Email', with 'Limpar pasta' selected. The main area is titled 'Programar itens' (Schedule items) and contains three task configuration panels. Each panel has a title, start/end time fields, and a repetition interval. The first panel is for '/Process\_Module/Process\_Schedule', the second for '/Process\_Module/Test', and the third for 'Enviar email'. The 'Enviar email' panel has a 'Configurações do calendário' (Calendar settings) section with various options and fields. Numbered callouts (1-6) point to specific UI elements: 1. 'Visualizar' button in the top navigation; 2. 'Test' process in the tree; 3. 'Hora de término' field in the first task panel; 4. 'Específico' tab in the second task panel; 5. 'Limpar pasta' operator in the sidebar; 6. 'Configurações do calendário' section in the third task panel.

Item:	Descrição:
1	<b>Editor de programação:</b> quando você abrir uma programação no Navegador da biblioteca, a caixa de diálogo Programação será exibida. Use a barra de ferramentas para ativar, configurar a validade, disponibilizar ou salvar a programação. Clique em qualquer uma das cinco guias, incluindo Editor de programação.
2	<b>Painel de processos:</b> selecione os processos que deseja incluir e arraste-os para a lista Programar itens.
3	<b>Processo programado:</b> defina a duração e a frequência para a execução do processo em um único dia, das 00h00 às 23h45.
4	<b>Propriedades do processo:</b> clique em Propriedades para exibir as propriedades do processo nas guias Geral e Específico.
5	<b>Painel Operadores:</b> selecione os operadores que deseja incluir e arraste-os para a lista Programar itens.
6	<b>Propriedades do operador:</b> clique em Propriedades para exibir as propriedades do operador nas guias Geral e Específico.

## Criar um objeto de programação

Para criar qualquer objeto de automação no CA Process Automation, consulte o tópico [Criar um objeto](#) (na página 67).

## Programar processos e tarefas do operador

Você pode configurar um processo ou operador para ser executado como uma única tarefa ou uma série de tarefas em uma programação. A diferença entre especificar um operador como uma tarefa em uma programação e especificar um operador em um processo é que o operador programado é iniciado em uma hora programada, em vez de como uma etapa em um processo. Você também pode programar qualquer processo para ser iniciado usando um operador Iniciar processo em uma programação.

### Siga estas etapas:

1. Na biblioteca, clique duas vezes em um objeto Programação.
2. No Editor de programação, identifique os processos e os operadores que você deseja programar.
3. Expanda o painel Processos ou Operadores e arraste um processo ou um operador disponível para a lista Programar itens. Você também pode clicar com o botão direito do mouse em um processo ou um operador e selecionar Adicionar.
4. Para cada item, preencha os seguintes campos:

#### hora de início

O horário de início para uma tarefa começar a executar em dias programados.

#### Intervalo de repetição (minutos)

Indica se uma tarefa será executada repetidamente entre os horários de início e término e, em caso positivo, com que frequência. Por exemplo, a cada 2 minutos, ou a cada 120 minutos (2 horas). Sempre que a tarefa é repetida, uma nova instância da tarefa é criada. Especifique o número de minutos de uma inicialização para a próxima no campo adjacente (minutos). Por exemplo, o valor 120 na caixa Minutos repete uma tarefa a cada duas horas.

Certifique-se de que a hora de término é mais recente do que a última vez em que deseja que a tarefa repita. Por exemplo, você configura uma tarefa da seguinte maneira:

- Hora de início 00h00
- Caixa de seleção Intervalo de repetição selecionada
- Campo (minutos) definido como 120 minutos
- Hora de término 16h00

Essas configurações programam o processo ou operador para ser iniciado pela primeira vez às 00h00, repetir a cada duas horas e ser executado pela última vez antes das 16h00.

#### Hora de término

Para uma tarefa repetida, o horário em que a tarefa interrompe a repetição em qualquer dia programado.

5. Neste ponto, você tem uma tarefa programada para repetir no intervalo especificado todos os dias a partir da hora de início especificada até a hora de término especificada. Continue com as etapas restantes apenas se você precisar fazer alterações detalhadas nas datas ou nas propriedades da programação.
6. Expanda um grupo de itens ou clique em Propriedades.  
As guias Geral e Específico são exibidas.
7. Na guia Geral:
  - a. Expanda Configurações do calendário e selecione os seguintes campos:
    - **Apenas seleção manual:** considera apenas as datas programadas manualmente. Quando nenhum calendário é especificado em um item programado, o item é considerado como programado diariamente, exceto quando essa opção é selecionada. Quando essa opção é selecionada, as datas de execução devem ser explicitamente programadas em Datas incluídas manualmente, Datas excluídas manualmente ou em ambas.
    - **Incluir calendário:** um objeto Calendário que programa datas para executar a tarefa.
    - **Excluir calendário:** um objeto Calendário de férias que especifica os dias fechados nos quais as tarefas não são executadas. O caminho completo do calendário usado para especificar os fechados dias (aqueles em que uma tarefa não pode ser programada). Não há dias fechados quando nenhum calendário de férias é especificado aqui.
    - **Dias por turno:** ativa as regras que alternam as datas em que as tarefas são executadas quando uma data especificada pelo objeto Calendário corresponde a uma data fechada. O número de dias para trocar uma data programada quando a data programada cai em uma data de fechamento. O turno não pode ser negativo ou zero. Quando esse valor é negativo a data muda para frente. Quando esse valor for zero, as datas fechadas são simplesmente ignoradas sem reprogramar a tarefa.
    - **Não há dias excluídos:** marque essa caixa de seleção para contar apenas os dias em aberto quando alternar a data programada a fim de evitar uma data fechada.
    - **Máximo de turnos:** quando uma tarefa é reprogramada porque a data programada original corresponde a um dia fechado, é possível que a nova data também corresponda a um dia fechado. Esse parâmetro define o número máximo de turnos que permitidos. Essa situação não ocorre quando a caixa de seleção Não há dias excluídos está marcada.
  - b. Expanda **Datas incluídas manualmente** para listar datas individuais a serem incluídas na programação.
  - c. Expanda **Datas excluídas manualmente** para listar datas individuais a serem excluídas, até mesmo de calendários de *inclusão* especificados anteriormente.
  - d. Expanda **Nome da tarefa** para inserir um nome mais significativo para a tarefa.

8. Clique na guia Específico e expanda os grupos de campos que variam por operador ou processo. Para um processo, os grupos Iniciar processo e Configurações de execução são exibidos. Para um operador, Configurações de execução e outros parâmetros são exibidos. Configure os campos.
9. Para definir o intervalo de datas válido para toda a programação, clique em Validade. Especifique uma data distante no futuro, como 31/12/2050, para continuar a avaliar as datas indefinidamente ou até que a programação seja desativada manualmente.
10. Para excluir uma tarefa, clique em X na borda direita.
11. Clique em Ativar para iniciar seus itens programados.
12. Clique em Disponibilizar e feche a caixa de diálogo Programação.
13. Monitore os itens programados na guia Operações.

## Visualizar todas as ocorrências de uma tarefa programada

É possível visualizar as ocorrências de tarefas programadas de um dia específico. Ao configurar uma tarefa para repetir em um dia específico, o aplicativo controla cada ocorrência da tarefa. Por exemplo, se uma tarefa for repetida a cada 10 minutos por meia hora, a visualização mostrará três ocorrências. Use esse procedimento para planejar para uma data futura ou para exibir os resultados de tarefas que foram programadas em uma data específica no passado.

### Siga estas etapas:

1. Na caixa de diálogo Programação, clique na guia Editor de programações para configurar a lista de itens programados para todo o período de validade.
2. Clique na guia Visualizar. Se a tarefa for repetida, várias ocorrências no mesmo dia são exibidas na guia Visualizar.

3. Na guia Visualizar:
  - a. No campo Preview Date, selecione uma data no intervalo válido para a programação.
  - b. Na lista suspensa Nós, selecione Todos os nós ou um touchpoint do orquestrador específico.
  - c. Selecione as opções Atual ou Arquivado.
  - d. Clique em Atualizar.

O aplicativo avalia as regras do calendário e o período de validade associado às tarefas antes de listar as ocorrências apropriadas.

- Ao selecionar uma data passada, a guia Visualizar inclui as tarefas que foram iniciadas, seu estado, a hora de início e a hora de término.
  - Ao selecionar a data atual, a guia Visualizar indica se as ocorrências foram iniciadas, a hora de início, o estado e a hora de término.
  - Ao selecionar uma data futura, a guia Visualizar inclui todas as ocorrências para a data selecionada se a programação estiver ativa no touchpoint selecionado.
4. Clique duas vezes em uma ocorrência para exibir um painel Propriedades somente leitura que mostra as definições de configuração para a tarefa.
  5. Clique na guia Editor de programações para fazer alterações na programação.
  6. Clique em Validade para definir o intervalo de datas válido para toda a programação.
  7. Clique em Ativar para iniciar seus itens programados.
  8. Clique em Disponibilizar e feche a caixa de diálogo Programação.
  9. Monitore os itens programados na guia Operações.

## Usando programações

Você deve ativar e disponibilizar uma programação para usá-la. É possível ativar uma programação no touchpoint do orquestrador específico em que ela reside. Quando uma programação está ativa em vários touchpoints, ela funciona como uma instância separada em cada touchpoint. Isso permite programar as mesmas tarefas em vários computadores simultaneamente. Exemplos desses tipos de tarefas incluem log de vínculo, instalações de software, atualizações e backups de arquivo.

**Observação:** quando você executar uma programação, a programação e qualquer operador nela usará apenas as cópias disponibilizadas dos objetos aos quais eles fazem referência.

## Monitorar programações ativas

Depois de ativar uma programação, é possível monitorá-la usando o link de Programações ativas na página Operações.

### Siga estas etapas:

1. Clique na guia Operações.
2. Na página Operações, expanda o painel Links.
3. Clique em Programações ativas.
4. Na barra de ferramentas, selecione um orquestrador e ambiente e, em seguida, clique em Atualizar.
5. Na tabela de Programações ativas, clique duas vezes em uma programação.

A caixa de diálogo Programação é exibida.

6. Na caixa de diálogo Programação:
  - a. Edite os itens programados. Consulte o tópico [Programar processos e tarefas do operador](#) (na página 401).
  - b. Na barra de ferramentas, clique em Ativar para ativar a programação.
  - c. Na barra de ferramentas, clique em Desativar para desativar a programação. Você também pode desativar uma programação na página Operações.

**Observação:** as permissões atribuídas determinam se você pode listar, abrir ou editar um objeto de programação específico. Um administrador de conteúdo ou o proprietário do objeto de automação pode alterar as permissões em um objeto de automação.

7. Clique em Disponibilizar ou em Salvar e Fechar.

## Monitorar todas as ocorrências de todas as tarefas programadas

É possível monitorar as ocorrências programadas de todas as tarefas de um dia específico. Ao configurar para que uma tarefa seja repetida várias vezes ao dia por vários dias, o CA Process Automation rastreia cada ocorrência da tarefa. Por exemplo, se uma tarefa for repetida a cada 10 minutos por meia hora (3 ocorrências) a cada dia durante 1 ano, o CA Process Automation rastreará 1.095 ocorrências. Use esse procedimento para planejar para uma data futura ou para exibir os resultados de todas as tarefas que foram programadas para uma data específica no passado.

### Siga estas etapas:

1. Clique na guia Operações.
2. Expanda o painel Links nas página Operações.
3. Clique em Programações globais.

4. Na barra de ferramentas, selecione um orquestrador e um ambiente e, em seguida, clique em Atualizar.
5. Na tabela de Programações globais:
  - a. No campo Preview Date, selecione uma data no intervalo válido para a programação.
  - b. Na lista suspensa Nós, selecione Todos os nós ou um touchpoint do orquestrador específico.
  - c. Selecione a opção Atual ou Arquivado.
  - d. Clique em Atualizar.

O CA Process Automation avalia as regras do calendário e o período de validade que estão associados às tarefas antes de exibir as ocorrências apropriadas na lista. Se a tarefa se repetir, várias ocorrências no mesmo dia serão exibidas.

- Ao selecionar uma data passada, a lista incluirá as tarefas que foram iniciadas, seu estado, a hora de início e a hora de término.
  - Ao selecionar a data atual, a lista indica se as ocorrências foram iniciadas, a hora de início, o estado e a hora de término.
  - Ao selecionar uma data futura, a lista inclui todas as ocorrências para a data selecionada se a programação estiver ativa no touchpoint selecionado.
6. Para exibir uma janela somente leitura Propriedades que mostra as definições de configuração para a tarefa programada, clique duas vezes na ocorrência apropriada.
  7. Na barra de ferramentas:
    - Clique em Conjunto de dados para exibir o conjunto de dados do processo ou do operador.
    - Clique em Cancelar ou Em espera para liberar ou pausar a ocorrência.
    - Clique em Redefinir para reiniciar uma ocorrência.

## Gerenciamento de tarefas

É possível gerenciar as tarefas do CA Process Automation criando formulários de interação de usuário para permitir que os usuários controlem as tarefas ou forneçam entradas personalizadas. Na Lista de tarefas da página Operações ou na lista Minhas tarefas da página Início, clique com o botão direito do mouse em uma tarefa para:

### Responder

Modifique o processo de alguma maneira usando um formulário da caixa de diálogo Responder. Por exemplo, você pode alterar os parâmetros ou os valores do campo antes de clicar em Concluir para concluir a tarefa.

### Assumir

Atribua temporariamente a você mesmo toda a propriedade e a responsabilidade da tarefa pendente.

### Retornar

Para tarefas com um status de apenas Assumido, libere sua propriedade exclusiva da tarefa. A tarefa é retornada ao seu responsável designado ou aos usuários ou grupos delegados.

### Usuário delegado

Atribua a tarefa a um usuário ou grupo secundário, conhecido como um usuário ou grupo delegado.

### Transferir

Atribua a tarefa a um responsável diferente.

### Abrir instância de processo

Exiba a tarefa no contexto da instância do processo pai e revise o design do processo.

### Atualizar

Atualize a lista de tarefas com as alterações mais recentes de todos os usuários e todas as atividades do sistema.

### Propriedades

Visualize mais informações sobre a tarefa, inclusive sua descrição, sua data de vencimento, seu status e seus responsáveis e delegados.

## Atribuir uma tarefa a um usuário

Para criar uma tarefa, crie um processo usando o operador *Atribuir tarefa de usuário* e, em seguida, inicie-a.

É possível especificar os seguintes atributos durante a criação de uma tarefa:

1. Abra um processo no Criador de processos.
2. Na paleta Operadores, expanda Controle de processo ou procure pelo operador Atribuir tarefa de usuário.
3. Arraste o operador Atribuir tarefa de usuário para o seu processo.
4. Clique duas vezes no operador Atribuir tarefa de usuário.
5. Na paleta Propriedades:
  - a. Expanda *Responsáveis* e insira os usuários e os grupos a serem atribuídos a essa tarefa.
  - b. Expanda *Transferir/delegar filtros* para permitir a delegação da tarefa, que é restrita a usuários ou grupos especificados.
  - c. Expanda *Tarefa de usuário* e preencha os seguintes campos:

**Título**

O título da tarefa.

**Descrição**

Uma descrição da tarefa.

**Formulário de solicitação de interação**

O caminho da biblioteca para o formulário de solicitação de interação.

**Código de inicialização de dados do formulário**

É possível escrever em JavaScript para preencher esse campo.

**Mostrar página de aprovação**

Uma caixa de seleção que especifica se uma tela de aprovação deve ser exibida no final do formulário de solicitação de interação. O usuário que está trabalhando na tarefa pode aprovar ou rejeitá-la usando a página de aprovação.

- d. Expanda os seguintes grupos de propriedades do operador comum para especificar qualquer informação do criador sobre o operador Atribuir tarefa de usuário:
    - Configurações de execução
    - Simulação
    - Informações
6. Na barra de ferramentas do Criador, clique em Salvar.

## Lista de tarefas

É possível trabalhar com tarefas na página inicial ou na página Operações. A página Início exibe apenas os campos Status, Título, Descrição e Data de vencimento na tabela conveniente *Minhas tarefas*. Use a Lista de tarefas da página Operações para exibir informações mais detalhadas sobre as tarefas.

As tarefas são provenientes de processos que incluem um operador Atribuir tarefa de usuário e um formulário de solicitação de interação. Você pode classificar a lista de tarefas em ordem crescente/decrescente clicando nos cabeçalhos da coluna. Use a coluna *Status* para determinar se uma tarefa está pendente, concluída, aprovada, rejeitada ou assumida.

Na página Operações, você pode filtrar a lista de tarefas para exibir o seguinte:

- Apenas as suas tarefas (*Minhas tarefas*, as tarefas que estão atribuídas ao usuário atual)
- Apenas as tarefas atribuídas a todos os *grupos* aos quais você pertence
- *Todas as tarefas*

A lista de tarefas na página Operações exibe as seguintes colunas dos campos para cada tarefa:

- ID da tarefa
- Título
- Descrição
- hora de início
- Data de vencimento
- Data de conclusão
- Status
- Responsáveis
- Usuários delegados

## Administrar disparadores

Você poderá controlar os processos com aplicativos externos usando qualquer um dos métodos a seguir:

- Gatilhos
- Serviços web (SOAP)
- Utilitário da linha de comando
- Scripts

Recomenda-se usar chamadas SOAP em vez de disparadores, pois os serviços web são mais robustos. Aplicativos que não podem fazer chamadas SOAP podem usar os disparadores como alternativa.

Os disparadores permitem que aplicativos externos iniciem um processo no CA Process Automation. Um disparador chama o processo do CA Process Automation que está definido no conteúdo XML ou em uma SNMP trap. O conteúdo XML pode ser entregue para o local do arquivo ou o endereço de email configurado. O conteúdo do SNMP trap pode ser enviado em um OID que corresponda a uma expressão regular configurada. O CA Process Automation escuta as SNMP traps de entrada na respectiva porta configurada (162 por padrão).

Sempre que você iniciar um processo, começar uma operação, como executar o operador Iniciar processo, ou usar um disparador ou uma chamada SOAP, você estará agindo em nome de um usuário ou de um proprietário. Para disparadores ou chamadas SOAP, as informações sobre o proprietário do conteúdo estão contidas na carga ou em mensagens. Essas informações determinam as versões dos objetos de automação executados:

- Se você reservar um processo e, em seguida, executá-lo, chamá-lo ou dispará-lo (você é o proprietário do conteúdo e iniciador), o CA Process Automation usará sua versão reservada particular.
- Caso contrário, o CA Process Automation usará as versões atuais dos objetos de automação. Isso inclui os processos que não estão reservados ou que estão reservados por outro usuário.

É possível executar e verificar sua própria versão reservada antes de disponibilizar os objetos novamente ou atualizá-los.

## Controle de processos por meio de um aplicativo externo com chamadas SOAP

O Orquestrador do CA Process Automation expõe serviços web que permitem que aplicativos externos iniciem e controlem processos em uma biblioteca. As chamadas SOAP exigem um XML válido. Os métodos e os parâmetros expostos dos serviços web são descritos no WSDL. Para obter detalhes, recupere o WSDL do URL de domínio apropriado, dependendo se o CA Process Automation oferece suporte à comunicação segura e é agrupado. Nos exemplos a seguir, *load\_balancer\_hostname* é o nome de host ou o endereço IP do balanceador de carga do Apache.

- Seguro e sem cluster:  
`https://<DomainOrchestrator_hostname>:8443/itpam/soap?wsdl`
- Desprotegido e sem cluster:  
`http://<DomainOrchestrator_hostname>:8080/itpam/soap?wsdl`
- Seguro e em cluster:  
`https://<load_balancer_hostname>:<Apache_secure_port>/itpam/soap?wsdl`  
**Observação:** a porta segura do Apache é normalmente a 443.
- Desprotegido e em cluster:  
`http://<load_balancer_hostname>:<Apache_unsecure_port>/itpam/soap?wsdl`  
**Observação:** a porta desprotegida do Apache é normalmente a 80.

Para scripts de exemplo que usam chamadas SOAP para que o Orquestrador do CA Process Automation inicie os processos, navegue até a seguinte pasta:

```
<install_dir>/server/c2o/.c2orepository/public/scripts/trigger
```

**Observação:** o caminho `<install_dir>` geralmente é `C:\Arquivos de programas\CA\PAM`.

A subpasta Java contém uma ferramenta baseada em Java e todos os recursos que ela requer para iniciar os processos do CA Process Automation remotamente com o uso do SOAP. O caminho para a subpasta Java é o seguinte:

```
<install_dir>/server/c2o/.c2orepository/public/scripts/trigger/java
```

## Como os disparadores de arquivos e emails funcionam

Este tópico fornece uma descrição da sequência de processamento dos acionadores. Ele usa disparadores de arquivo e email como exemplos.

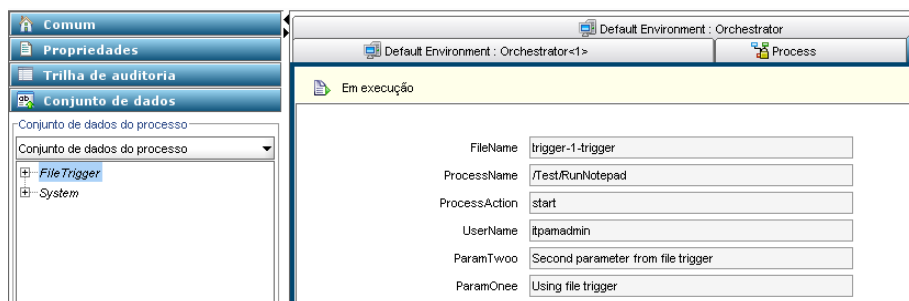
**Exemplo de acionador de arquivo e de email:**

1. Na frequência configurada, o CA Process Automation pesquisa o novo conteúdo na pasta e na conta de email configuradas.
2. Se um novo objeto de arquivo ou de email for encontrado, o CA Process Automation tentará executar o processo com base no conteúdo XML.

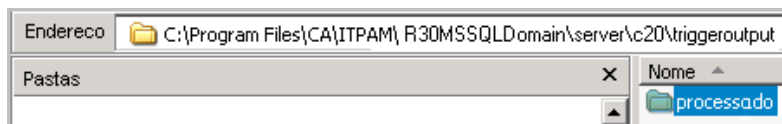
Uma ilustração do conteúdo XML válido para o disparador de arquivo é a seguinte:

```
<c2oflow version="1.0">
  <flow name="/Test/RunNotepad" action="start"> <!-- Full path of the process -->
  <auth>
    <user>itpamadmin</user> <!-- ITPAM Username -->
    <password>itpamdemo</password> <!-- ITPAM Password -->
  </auth>
  <options>
    <!-- Optional parameters for delayed execution
    <startDate>/startDate>
    <startTime>/startTime>
    -->
  </options>
  <params>
    <!-- Process initialization parameters, if needed -->
    <param name="ParamOne">Using file trigger</param>
    <param name="ParamTwo">Second parameter from file trigger</param>
  </params>
  </flow>
</c2oflow>
```

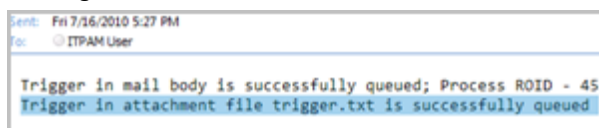
3. Quando o disparador executa o processo especificado em sua instância, o conjunto de dados do processo é preenchido com os valores contidos no XML. O exemplo a seguir demonstra como os valores no conteúdo XML são usados para preencher o conjunto de dados do processo do acionador de arquivo.



4. Os resultados são publicados na pasta de processados no caminho configurado.



5. Os emails recebidos na conta de email configurado são processados da mesma forma que o conteúdo XML recebido nos arquivos. Além disso, os emails de confirmação são enviados, em que o conteúdo determina se o conteúdo XML foi encontrado no corpo do email ou em um anexo. O exemplo a seguir mostra as duas mensagens:



## Monitorar o Conjunto de dados FileTrigger de um processo iniciado por um disparador de arquivo

Quando um conteúdo XML válido de um arquivo disparar uma instância de processo, você poderá monitorar o conjunto de dados do processo do disparador de arquivo na Exibição de processos. Use as descrições de campo abaixo para interpretar os valores exibidos.

### FileName

O nome do arquivo com o conteúdo que disparou o processo.

### <additional\_parameters>

Parâmetros adicionais que são passados sob o tag <params> ao disparar o conteúdo XML do arquivo.

## Monitorar o Conjunto de dados SMTP de um processo iniciado por um disparador de email

Quando um email com conteúdo XML válido disparar uma instância de processo, você poderá monitorar o conjunto de dados do processo SMTP com a Exibição de processos. Use as descrições de campo abaixo para interpretar os valores exibidos.

### SenderAddress

O endereço de email da conta da qual o email de disparo é enviado.

### SentDate

Exibe a data e a hora em que o email foi enviado.

### ReceivedDate

A data e a hora em que o servidor recebeu o email.

### MailSubject

Assunto do email de disparo.

### MessageNumber

Número da mensagem do email de disparo no momento em que o processo foi disparado.

**Observação:** esse número poderá ser alterado para o mesmo email, se as mensagens forem excluídas ou movidas da Caixa de entrada.

### MessageID

ID exclusiva do email no servidor.

### MailBody

O corpo da mensagem de email nestes casos:

- Quando um conteúdo XML válido em um anexo dispara o processo.
- Quando o processo do disparador padrão é iniciado, ou seja, quando nenhum conteúdo XML válido é encontrado no corpo do email ou no anexo.

**Observação:** esse valor de sequência de caracteres está truncado para os primeiros 64K caracteres no corpo do email.

### MailAttachments

Uma variável do tipo de matriz ValueMap que contém ValueMaps com as informações a seguir sobre os anexos:

- a. contentType: tipo de conteúdo do anexo.
- b. contentID: contentID do anexo, se houver.
- c. fileURL: URL por meio do qual o anexo pode ser exibido ou baixado.
- d. name: nome do anexo.
- e. attachmentID: ID exclusiva do anexo. Essa ID pode ser passada para as funções do sistema do JavaScript.

**Observação:** consulte a *Referência do criador de conteúdo* para obter detalhes sobre as funções do sistema do JavaScript.

### <additional parameters>

Esses parâmetros, passados sob a tag <params> no conteúdo XML de disparo, existirão apenas quando o conteúdo XML válido no corpo do email ou no anexo iniciar o processo do CA Process Automation.

## Formato do conteúdo XML para os disparadores de arquivo e email

Os aplicativos externos que usam disparadores de arquivo ou email para iniciar processos do CA Process Automation devem criar a entrada em um formato XML válido. O conteúdo XML pode ser gravado no corpo de um email ou enviado como um anexo. Se o XML for copiado para o corpo do email, ele não poderá conter mais do que o necessário para disparar um processo. Para disparadores de arquivo, o arquivo de disparo deve incluir todo o conteúdo.

Um exemplo de um formato XML válido é o seguinte:

```
<c2oflow version="1.0">
  <flow name="/Test/RunNotepad" action="start"> <!-- Full path of the process -->
    CA AuthMinder
      <user>pamadmin</user>          <!-- CA Process Automation Username -->
      <password>pamadmin</password> <!-- CA Process Automation Password -->
    </auth>
    <options>                        <!-- Optional parameters for delayed execution -->
      <startDate></startDate> <!--Start Date in [MM/dd/yyyy] format -->
      <startTime></startTime> <!-- Start Time in [HH:mm] format; HH in 24 hrs -->
    </options>
    <params>                          <!-- Process initialization parameters, if needed -->
      <param name="ParamOne">Uso do disparador de arquivo</param>
      <param name="ParamTwo">Segundo parâmetro do disparador de arquivo</param>
    </params>
  </flow>
</c2oflow>
```

## Considerações sobre a entrada da SNMP trap

O CA Process Automation oferece suporte aos SNMPv1 e SNMPv2 traps; entretanto, ele não processa os SNMPv3 traps. Quando um dispositivo de rede ou um aplicativo corporativo enviar uma SNMP trapv1 ou SNMPv2 que o CA Process Automation detecta na porta configurada, o CA Process Automation processa o conteúdo.

## Alterar a porta de escuta de SNMP Traps

Por padrão, o CA Process Automation escuta na porta 162 as SNMP traps projetadas para iniciar processos do CA Process Automation. Se você fechou a porta 162 no seu site e configurou uma alternativa, altere a configuração do CA Process Automation para essa porta no arquivo OasisConfig.properties. Em seguida, reinicie o serviço do orquestrador.

É possível alterar a porta em que o CA Process Automation escuta as SNMP traps.

### Siga estas etapas:

1. Faça o logon no servidor em que o Orquestrador de domínio está configurado.
2. Vá até a seguinte pasta ou diretório:  
`dir_instalação/server/c2o/.config/`
3. Abra o arquivo OasisConfig.properties
4. Altere o valor na seguinte linha de 162 até o número da porta que você está usando para SNMP traps.  
`oasis.snmptrigger.service.port=162`
5. Salve o arquivo.
6. Reinicie o serviço do orquestrador.
  - a. Interrompa o orquestrador.
  - b. Inicie o orquestrador.

Assim que o serviço for reiniciado, o CA Process Automation começará a escutar na porta configurada. O CA Process Automation escuta novos SNMP traps que atendam aos critérios configurados no disparador de SNMP.

## Monitorar o Conjunto de dados SNMP de um processo iniciado por um disparador de SNMP trap

Quando um SNMP trap disparar uma instância de processo, você poderá monitorar o conjunto de dados do processo SNMP com a Exibição de processos. Use as descrições de campo abaixo para interpretar os valores exibidos.

**SenderAddress**

Endereço IP da origem.

**AgentIPAddress**

Endereço IP do agente SNMP, se disponível no trap.

**SNMPVersion**

Versão de SNMP trap.

**ErrorIndex**

Índice do erro de trap

**AgentUptime**

Tempo de atividade do agente que envia o trap.

**EnterpriseOID**

O OID (object identifier - identificador de objeto) do objeto gerenciado que gerou o SNMP trap.

**PayloadOIDs**

IDs de objeto presentes na carga da trap. As IDs de objeto de carga representam uma variável da matriz de sequência de caracteres do CA Process Automation.

**PayloadValues**

Valores na carga que correspondem aos valores em *PayloadOIDs*. Esses dados também são uma variável da matriz de sequência de caracteres do CA Process Automation.

**Observação:** se houver vários filtros, a primeira correspondência será processada.



# Capítulo 10: Executando, testando e depurando processos

---

Este capítulo descreve como executar, testar e depurar processos interativamente durante o desenvolvimento. Os mesmos métodos podem ser usados para executar processos em um ambiente de produção.

Quando desejar executar um processo, você poderá iniciá-lo usando qualquer um dos métodos a seguir:

- Início de processo manual
- Formulário de solicitação inicial
- Chamada de outros processos
- Disparador usando aplicativos externos, FTP, chamadas SOAP, interceptações SNMP ou mensagens SMTP (email).

O módulo de fluxo de trabalho em um orquestrador executa os processos. Quando você inicia um processo em um orquestrador, o módulo de fluxo de trabalho cria e executa uma cópia do objeto de processo na biblioteca de automação do orquestrador. Esta cópia em execução é uma *instância* do processo. O módulo de fluxo de trabalho cria uma instância separada de um processo toda vez que você inicia um processo (ou outro processo ou aplicativo inicia um processo).

É possível abrir, exibir e trabalhar com uma instância de um processo durante ou após a conclusão de sua execução. As alterações feitas em uma instância de um processo afetam apenas essa instância, e não o objeto de processo original, armazenado em uma biblioteca de automação.

Quando um processo é iniciado, ele conecta-se aos módulos corretos de agente ou orquestrador nos computadores de rede gerenciados. Um processo executa sua funcionalidade de operador designada, testa condições e pratica dependências. Quando condições de erro ocorrerem, um processo executará ações corretivas e notificará os operadores e administradores, quando necessário. Um administrador pode usar o Monitor de aplicativo para monitorar os processos em execução e executar ações corretivas.

Esta seção contém os seguintes tópicos:

[A página Operações](#) (na página 421)

[Regras de execução](#) (na página 427)

[Segurança de tempo de execução](#) (na página 428)

[Tratamento de exceção](#) (na página 431)

[Executar processos de forma interativa](#) (na página 435)

[Estados do processo](#) (na página 439)

[Depurar um processo](#) (na página 439)

[Controlar a ramificação de um processo](#) (na página 443)

[Simular o processamento dos operadores](#) (na página 446)

## A página Operações

A página Operações tem as seguintes funções:

- Monitora as tarefas selecionadas e os elementos do sistema gerenciados pelo CA Process Automation
- Exibe as propriedades de pacotes de conteúdo importados na paleta Pacotes de conteúdo
- Exibe todas as atividades do processo para uma biblioteca de automação selecionada, incluindo:
  - Instâncias do processo
  - Programações
  - Invocações de módulos
  - Conjuntos de dados
  - Prompt do usuário
  - Recursos

The screenshot displays the CA Process Automation 'Operations' page. The interface includes a navigation menu on the left, a central bar chart showing process instance counts across various states, and a table listing individual process instances. Numbered callouts (1-5) highlight specific UI elements:

- 1: Operations tab in the top navigation bar.
- 2: Start Process button in the left navigation menu.
- 3: Failed bar in the Process Instances bar chart.
- 4: A row in the Process Instances table.
- 5: A tooltip for the Failed bar in the chart.

Instance	State	Start Time	End Time	User	Touchpoint	Content Package Name	Content Package Release Version
start_system_process_1261	Completed	2012/01/19 11:10:59 AM	2012/01/19 11:12:0...				
start_script_1255	Completed	2012/01/19 11:10:59 AM	2012/01/19 11:11:5...				
Driver_338	Running	2012/01/19 10:52:00 AM					
UCF_Test_CRUD_791	Completed	2012/01/19 10:54:01 AM	2012/01/19 10:54:0...				
TFTP_1409	Failed	2012/01/19 11:21:55 AM	2012/01/19 11:22:5...				
SNMPTraps_TrapTypes_1234	Completed	2012/01/19 11:09:34 AM	2012/01/19 11:10:2...				
SNMPTraps_PayloadTypes_1208	Completed	2012/01/19 11:09:16 AM	2012/01/19 11:09:3...				
SNMPv4_1247	Completed	2012/01/19 11:10:36 AM	2012/01/19 11:10:4...				
SNMPGet_SystemInformation_1193	Completed	2012/01/19 11:08:31 AM	2012/01/19 11:09:1...				
Select_Expression_1204	Completed	2012/01/19 11:11:14 AM	2012/01/19 11:12:2...				
Select_Inline_Negative_1303	Completed	2012/01/19 11:11:55 AM	2012/01/19 11:12:2...				
Select_Expression_Negative_1293	Completed	2012/01/19 11:11:21 AM	2012/01/19 11:11:5...				
renamefolder_959	Completed	2012/01/19 11:00:19 AM	2012/01/19 11:00:2...				
Process_Simulation_705	Blocked	2012/01/19 10:52:13 AM					
Proc_JDBC_List_View_1395	Completed	2012/01/19 11:14:20 AM	2012/01/19 11:14:3...				
Proc_Cust_View_Tables_1477	Completed	2012/01/19 11:16:33 AM	2012/01/19 11:16:4...				
Proc_Cust_List_View_1471	Completed	2012/01/19 11:16:15 AM	2012/01/19 11:16:3...				
Proc_Cust_List_UsedSpace_1494	Completed	2012/01/19 11:17:10 AM	2012/01/19 11:17:2...				
Process_278	Blocked	2012/01/19 09:51:00 AM					
Process_272	Blocked	2012/01/19 09:50:00 AM					
Process_27_55	Completed	2012/01/19 09:08:05 AM	2012/01/19 09:08:05				

Item: Descrição:

Item:	Descrição:
1	<b>Guia Operações:</b> clique nessa guia para navegar para a página Operações, um painel de automação de alto nível. Na barra de ferramentas, é possível selecionar um orquestrador, abrir uma instância do processo e controlá-la.
2	<b>Painel Links:</b> selecione um link para exibir os itens associados. Por exemplo, clique em Instâncias de processo para exibir todos os processos por estado. Expanda um grupo para procurar as pastas e selecionar uma instância. Alguns objetos incluem um menu de atalho de comandos que você pode acessar clicando com o botão direito do mouse. Por exemplo, a ilustração mostra que o usuário clicou com o botão direito do mouse em um subprocesso de um objeto Exibição de processos e clicou em Iniciar processo no menu de atalho.
3	<b>Área do Gráfico:</b> aponta para uma barra na área do Gráfico para exibir o número de itens que correspondem ao estado associado. Clique em uma barra para exibir somente os itens que correspondem ao estado associado.
4	<b>Tabela ou lista de operações:</b> de acordo com a sua seleção no painel Links, essa área exibe os dados resultantes em uma tabela ou lista.

## Filtros para instâncias do processo

É possível aplicar os seguintes filtros a um atalho do gerente de touchpoint em um objeto de exibição de processos para definir os objetos exibidos pelo atalho:

### Todas as instâncias

Exibe todas as instâncias de processo no orquestrador atual. Todas as instâncias são exibidas apenas se todos os filtros estiverem desmarcados ou se todos os filtros estiverem selecionados.

### Em fila

Exibe todas as instâncias que estão no estado Em fila e aguardando para serem executadas.

### Em execução

Exibe instâncias do processo em execução no touchpoint. Este filtro não lista as instâncias suspensas nos estados Aguardando, Suspenso ou Ponto de interrupção.

### Suspenso

Exibe as instâncias do processo que um usuário ou o aplicativo no touchpoint suspenderam atualmente.

### Aguardando

Exibe as instâncias do processo que possuem um estado de execução inativo, como processos em que todos os operadores ativos estão aguardando por um evento assíncrono externo. O evento poderia ser uma interação do usuário, a data e a hora de destino ou outra operação de execução longa.

**Bloqueado**

Exibe as instâncias de processo que estão bloqueadas porque não há nenhum caminho disponível. Uma instância do processo bloqueada poderá ser concluída depois que um usuário suspender o processo e fornecer um caminho válido. O estado bloqueado reflete as seguintes situações:

- A instância do processo exige a intervenção do usuário.
- Uma instância do processo não pode continuar devido a uma condição inesperada.

**concluído**

Exibe todas as instâncias de processo no touchpoint que foram concluídas sem problemas.

**Com falha**

Exibe todas as instâncias de processo no touchpoint que foram encerradas antes da conclusão.

**Cancelado**

Exibe todas as instâncias de processo no touchpoint que foram encerradas de forma anormal. Para identificar os processos com problemas rapidamente, examine esta pasta. Para solucionar o problema, abra a instância que falhou.

## Filtrar objetos exibidos por um atalho

A Exibição de processos inclui um atalho para cada um dos filtros. É possível usar o procedimento a seguir para adicionar filtros à Exibição de processos.

**Siga estas etapas:**

1. Abrir o objeto Exibição de processos: selecione uma pasta Biblioteca e clique duas vezes no objeto Exibição de processos.
2. Na guia Exibição de processos, selecione um processo a partir de uma pasta Biblioteca no painel esquerdo e, em seguida, clique em Adicionar Novo na barra de ferramentas.  
O objeto selecionado é adicionado como um atalho na Exibição de processos.
3. Expanda o atalho do objeto do processo atalho e selecione os filtros.
4. Salve as mudanças.

Para exibir a instância do objeto para os filtros selecionados, clique na guia Operações e expanda a paleta Exibição de processos. É possível selecionar o novo atalho a partir do local em que foi criado o atalho. Instâncias correspondentes são exibidas para o objeto do processo, dependendo dos filtros especificados.

## Objetos Exibição de processos

Um objeto Exibição de processos fornece um modo fácil para monitorar o status de outros objetos de automação. Você cria objetos Exibição de processos no Navegador da biblioteca. É possível adicionar objetos (tecnicamente ponteiros para objetos) de vários domínios, orquestradores, touchpoints e outras bibliotecas. Quando você exibir o objeto Exibição de processos na página Operações, ele exibirá o estado atual dos objetos monitorados.

### Criar um novo objeto Exibição de processos

Defina um novo objeto Exibição de processos no Navegador da biblioteca e monitore o status de cada objeto que ele inclui na página Operações.

**Observação:** para definir direitos e executar outras tarefas de objeto, a prática recomendada é organizar os objetos em pastas. Não crie objetos no nível raiz, pois não é possível gerenciá-los como um grupo.

#### Siga estas etapas:

1. Clique na guia Biblioteca.
2. No painel Pastas da biblioteca
  - a. Selecione um orquestrador no ambiente de produção ou criação.
  - b. Selecione uma pasta.
3. Na barra de ferramentas, clique em Novo e selecione Exibição de processos.  
Um novo objeto de exibição de processos é exibido na pasta selecionada.
4. No campo Nome, digite um nome para a exibição de processos.
5. Clique duas vezes na nova exibição de processos.  
A Exibição de processos é aberta em uma janela do navegador.
6. No painel à esquerda
  - a. Selecione o domínio, ambiente, orquestrador, touchpoint, grupo de hosts e a pasta para o objeto que você deseja adicionar à exibição de processos.
  - b. Selecione o objeto que deseja incluir na exibição de processos.
7. Na barra de ferramentas, clique em Adicionar novo.  
O objeto selecionado é adicionado à exibição de processos. O nome, o tipo de objeto, o caminho de referência, o modo e a descrição do objeto selecionado são exibidos.
8. Se você tiver adicionado um processo, selecione um ou mais estados que deseja mostrar na exibição de processos, por exemplo, em fila, suspenso e bloqueado.
9. Clique em Salvar.

## Monitorar objetos de dentro de uma Exibição de processos

É possível abrir uma instância de um processo mostrado no painel Detalhes da janela Exibição de processos. Abrir uma instância do processo permite monitorar ou editar de perto a execução da instância do objeto. As alterações a uma instância do objeto afetam apenas a execução dessa instância; elas não alteram a definição do objeto na Biblioteca.

**Observação:** um usuário do CA Process Automation deve ter permissões suficientes para exibir ou editar um objeto na janela Exibição de processos.

Para salvar as alterações em uma instância do objeto, clique com o botão direito do mouse na instância do processo em Exibição de processos e clique em Exportar.

**Observação:** para obter mais informações sobre como trabalhar com instâncias do processo na Exibição de processos, consulte o *Guia do administrador de conteúdo*.

## Suporte ao caminho relativo estendido

Uma solução de automação desenvolvida no CA Process Automation consiste em vários tipos de objetos de automação que podem incluir processos, conjuntos de dados, formulários de solicitação inicial e formulários de solicitação de interação.

O CA Process Automation permite que os usuários movam uma solução de automação de qualquer nível em toda a hierarquia da biblioteca. Por exemplo, os usuários podem mover objetos de uma pasta para outra ou podem mover os objetos de automação para uma biblioteca de domínio diferente. Quando mover as soluções de automação através de uma biblioteca, o caminho raiz dos objetos muda. Para mover uma solução de automação sem quebrar ou alterar a relação entre os objetos apesar da mudança na pasta raiz, use a opção Caminho relativo estendido.

**Observação:** o CA Process Automation fornece as opções Relativo e Absoluto na lista suspensa da coluna MODE do atalho do objeto adicionado na Exibição de processos e no Pacote.

## Objetos de pacote de conteúdo

O objeto de automação de pacote de conteúdo agrupa os objetos do CA Process Automation que o criador de conteúdo exportou para um ambiente diferente. O criador de conteúdo adiciona os objetos de automação em uma pasta e exporta a pasta como pacote de conteúdo. Quando você exibir o objeto de pacote de conteúdo na página Operações, ele exibirá o estado atual dos objetos importados.

### Mais informações

[Objetos de pacote de conteúdo](#) (na página 463)

## Monitorar objetos de dentro de uma Exibição de processos

É possível abrir uma instância de um objeto no painel Detalhes da janela Pacotes de conteúdo. A abertura de um objeto permite monitorar de perto a instância do objeto enquanto a instância é executada. As alterações em uma instância de objeto afetam apenas a maneira como essa instância é executada; as alterações não modificam a definição do objeto na biblioteca.

**Observação:** um usuário do CA Process Automation deve ter permissões suficientes para exibir ou editar um objeto na janela Pacote de conteúdo.

É possível executar as seguintes ações em um objeto em um pacote de conteúdo:

- Iniciar ou suspender um processo
- Iniciar um formulário de solicitação inicial
- Tornar um operador personalizado disponível ou indisponível
- Ativar ou desativar uma programação

## Regras de execução

Os operadores em um processo podem ter vários links de entrada e saída. Um link de entrada serve como uma ordem de execução e chama o operador. Cada link de saída corresponde a um determinado resultado do operador. Todos os operadores têm links de saída predefinidos (como Cancelado, Concluído, Falha ou Com êxito). Alguns operadores também permitem usar uma expressão booleana para definir um link de saída personalizado com base nos resultados e o valor das variáveis acessíveis aos operadores em um processo.

As regras de execução de um processo são as seguintes:

- Os operadores iniciais no fluxo principal podem ter um link de entrada. Se tiver um link de entrada, ele não poderá ter um link de saída e agir como um operador Redefinir (por exemplo, usado para redefinir um processo concluído).
- Os operadores Interromper não têm saída e concluem a execução de um processo.
- Todos os links de saída com condições booleanas avaliadas como Verdadeiro estão ativados e levam à ativação dos operadores de saída subsequentes. Links de saída padrões são mutuamente exclusivos. Todos os links personalizados para os quais a expressão booleana for avaliada como verdadeira estão ativados e levarão aos operadores de saída subsequentes.
- Os operadores (exceto operadores recapitalizados em uma ramificação em loop de um processo) são processados somente uma vez durante a execução de uma ramificação de um processo. Quando um link de um operador concluído leva a um operador que já foi ativado, o operador ativado não é processado uma segunda vez. Após a ativação, o link processado está indisponível para o processamento subsequente de um processo.

Alguns operadores suportam o processamento em loop, no qual o módulo de fluxo de trabalho executa o operador um número de vezes especificado ou indefinidamente. As condições de saída e os links de conexão do operador são avaliados apenas quando o loop é encerrado. O operador Loop permite também aplicar o processamento em loop e suas condições de saída a uma sequência embutida de operadores.

- Quebrar vínculos interrompe a execução de um loop em operadores que oferecem suporte ao uso do processamento em loop.

## Segurança de tempo de execução

O recurso opcional *Segurança de tempo de execução*, quando ativado, ajuda a verificar a identidade do usuário que está executando o processo ou a programação segura. O usuário de qualquer processo é o proprietário ou quem está chamando o processo. O usuário de qualquer programação é sempre o proprietário. A identidade do *usuário chamador* é a identidade do usuário que inicia um processo, uma programação ou um operador.

A imposição da Segurança de tempo de execução é usada quando um processo é iniciado, independentemente de como o processo for chamado. Por exemplo, a imposição da Segurança de tempo de execução será aplicada aos processos filho iniciados por processos pai.

Consulte [Especificar as propriedades de Segurança de tempo de execução](#) (na página 75).

Você também pode configurar um operador em um processo para ser executado no contexto do usuário que o chamou marcando a caixa de seleção *Executar como usuário chamador* listada sob Configurações de execução. Marcar essa opção indica que você deseja que o operador seja executado como se o usuário que iniciou o processo estivesse no controle. As configurações do operador substituem as configurações de propriedade de processo, se houver diferença.

## As propriedades afetam a segurança dos processos em execução

Somente o proprietário do processo ou os administradores do conteúdo do ambiente podem definir a opção *Segurança de tempo de execução*. Duas propriedades de processo afetam a segurança de tempo de execução das instâncias desse processo:

- Segurança de tempo de execução
- Executar como proprietário

### Segurança de tempo de execução

Especifica se é necessário aplicar a segurança de tempo de execução para este processo. A segurança de tempo de execução pode ser ativada ou desativada explicitamente ou por meio de herança. Quando for definida explicitamente, as alterações nas configurações herdadas não causarão impacto.

#### Herdar do orquestrador

Aplica a mesma configuração que está atualmente definida no orquestrador. A opção *Ativar segurança de tempo de execução* pode ser marcada ou desmarcada na guia Diretivas do orquestrador pai.

#### Ativar

Indica que você deseja ativar a *Segurança de tempo de execução*. Quando um usuário tentar iniciar uma instância desse processo, o CA Process Automation examinará a configuração de *Executar como proprietário* para o usuário.

- Se a opção *Executar como proprietário* estiver selecionada, o CA Process Automation determinará o usuário definido no momento como proprietário e iniciará o processo usando a identidade do proprietário. Se esse processo chamar um outro processo, esse processo será executado usando a identidade do proprietário do processo pai.

**Observação:** essa configuração poderá ser substituída no nível de operador se a opção *Executar como usuário chamador* estiver selecionada.

- Se a opção *Executar como Proprietário* estiver desmarcada, o CA Process Automation examinará as permissões para o usuário que está tentando iniciar uma instância do processo. Se o usuário tiver direitos de iniciar, o CA Process Automation permitirá que a instância do processo inicie com a identidade do usuário chamador. Se esse processo chamar um outro processo definido como usuário chamador, o CA Process Automation verificará os direitos de iniciar o processo filho.

#### Desativar

Indica que você deseja desativar a *Segurança de tempo de execução*. A caixa de seleção *Executar como proprietário* está desativada.

### Executar como proprietário

Essa caixa de seleção somente estará ativada se a opção *Segurança de tempo de execução* estiver ativada explicitamente ou através de herança.

#### Selecionado

Especifica que todas as instâncias do processo atual possam ser executadas com a identidade do proprietário (executar como proprietário). Quando um usuário autorizado iniciar o processo, o proprietário terá acesso aos processos filho e a outros objetos. O acesso pelo proprietário pode incluir objetos que o usuário chamador, que iniciou a instância, não tem permissão para acessar. Somente o proprietário do processo ou o administrador do conteúdo do ambiente pode definir essa propriedade.

#### Desmarcado

Especifica que a permissão Iniciar seja verificada em tempo de execução para o usuário chamador que tenta iniciar a instância do processo.

## Diretrizes para definir a Segurança de tempo de execução para um processo

Ao inicializar, uma instância de processo pode assumir uma das seguintes identidades:

- O usuário chamador, isto é, o usuário que iniciou a instância do processo.
- O proprietário do processo.

Ao configurar a segurança de tempo de execução no nível de processo, considere as seguintes diretrizes.

Seu objetivo:	Configuração necessária:
Execute o processo como usuário chamador. Aplique direitos de segurança de tempo de execução com a identidade do usuário que inicia a instância do processo.	<ul style="list-style-type: none"><li>■ Selecione <i>Ativar</i> no campo Segurança de tempo de execução.</li><li>■ Desmarque <i>Executar como proprietário</i>.</li></ul>
Execute o processo como proprietário. Aplique a segurança de tempo de execução executando as instâncias de processo com a identidade do proprietário, independentemente de quem as iniciar.	<ul style="list-style-type: none"><li>■ Selecione <i>Ativar</i> no campo Segurança de tempo de execução.</li><li>■ Selecione <i>Executar como proprietário</i>.</li></ul>
Desative a validação e a aplicação da propriedade do processo em tempo de execução.	Selecione <i>Desativar</i> no campo Segurança de tempo de execução. Essa opção ajuda a garantir a compatibilidade com as versões anteriores dos processos existentes.

### Mais informações:

[Determinar quando selecionar Executar como proprietário](#) (na página 74)

## Tratamento de exceção

A manipulação de exceções permite que você defina as sequências de operadores para exceções predefinidas em operadores de um processo, como Falha, Cancelar ou Resultado inesperado. Também é possível criar uma sequência padrão de operadores para executar qualquer exceção que esteja sem uma sequência explícita. Enquanto o módulo de fluxo de trabalho processa uma exceção, ele pausa a execução de quaisquer outros operadores no processo.

A manipulação de exceções usa prioridades ao avaliar as condições de saída em um operador. A tabela a seguir lista os tipos de exceção:

Prioridade	Tipo de exceção	Ocorre quando
1	Exceção do sistema	Há um nome incorreto de touchpoint, um agente inacessível ou qualquer tipo de falha de comunicação.
2	Resposta não identificada	Não há nenhum link de saída para uma determinada condição de saída.
3	Cancelado	Um operador é cancelado ou um usuário cancela um operador.
4	Tempo de expiração	Uma operação expira e não há caminho definido na porta de tempo limite para o fluxo principal.

Quando um operador de processo apresenta uma exceção, o módulo de fluxo de trabalho executa as seguintes ações:

- Suspende o processamento do processo depois de executar os operadores atuais.
- Tenta corresponder e executa uma exceção na seguinte ordem:

Prioridade	Correspondências	Ação
1	O manipulador de exceção definido no objeto de processo.	Executa o manipulador de exceção definido no objeto de processo.
2	O manipulador de exceções definido no objeto de processo padrão do orquestrador em execução no módulo de fluxo de trabalho.	Executa o manipulador de exceção definido no objeto de processo padrão para o orquestrador.
3	Nenhuma	Ignora a exceção. O módulo de fluxo de trabalho continua executando o processo.

## Criar manipuladores de exceção

A guia Manipulador de exceções permite criar sequências de operadores para as seguintes exceções predefinidas em um processo:

### Cancelado

Ocorre em um cancelamento especificado pelo usuário ou pelo operador.

### Erro do sistema

Ocorre com qualquer tipo de falha de comunicação. Por exemplo, quando o processo contém um nome de touchpoint incorreto ou se refere a um agente que não esteja em execução.

### Tempo limite

Ocorre quando estas duas circunstâncias são verdadeiras:

- O tempo limite do operador expira antes de ele concluir.
- O operador é configurado para tomar o caminho para tempo limite e terminar ou continuar com um resultado. O manipulador de exceções define o caminho do tempo limite.

### Resposta não identificada

Ocorre quando nenhum conector de saída corresponde à resposta.

### Siga estas etapas:

1. Clique na guia Criador.
2. Clique em Abrir, vá até a pasta com o processo a ser aberto, selecione o processo e clique em Abrir.
3. Selecione a guia Manipulador de exceções.
4. Expanda a pasta Padrão na paleta Operadores e arraste o operador Exceção para o processo.
5. Expanda a caixa de diálogo Propriedades do operador Exceção.
6. Expanda Informações e digite um nome no campo Nome.  
**Observação:** a prática recomendada é nomear os operadores em um manipulador de exceções padrão com um prefixo, para que eles não correspondam aos nomes de operadores no processo que carrega o manipulador de exceções padrão.
7. Ocorreu uma exceção de expansão.
8. Selecione um tipo de exceção na lista suspensa.
9. Na paleta, arraste mais operadores até o processo que conclui a regra para a exceção. Vincule os operadores na sequência de execução.

**Observação:** se você finalizar a sequência sem adicionar nenhum operador Interromper, o processo principal será retomado. Como alternativa, você pode parar o processo para um ou mais caminhos no Manipulador de exceções.

10. Na barra de ferramentas, clique em Salvar.  
A nova regra de exceção é adicionada.

Como parte da manipulação de exceção, é possível redefinir o operador e continuar o processo. Também é possível optar por ignorar uma exceção e continuar o processo. Para ignorar uma exceção, defina o operador no modo de simulação e continue o processo. O operador Redefinir reside na paleta Comum. É possível usar o operador Redefinir no painel do processo, no manipulador de exceções e no manipulador de mudança de rota.

### **Siga estas etapas:**

1. Clique com o botão direito do mouse em Adicionar, Redefinir para adicionar um operador Redefinir.  
  
A caixa de texto exibe uma entrada com uma lista suspensa para selecionar um dos operadores no processo atual. É possível adicionar vários nomes de operador.
2. Para manipular um nome de operador, clique em Excluir, Mover para cima e Mover para baixo, conforme apropriado.
3. Insira uma expressão que resolva para uma sequência de caracteres (por exemplo, nome de operador) ou uma lista de valores (por exemplo, nomes de operadores) em tempo de execução. Execute esta ação, em vez de selecionar um nome de operador na lista suspensa.
4. Marque ou desmarque a opção Continuar com resultado.

### **Selecionado:**

Torna a opção EndCondition disponível para selecionar Bem-sucedido ou Sem êxito.

#### **Bem-sucedido:**

Se uma condição de erro for atendida em tempo de execução, o CA Process Automation presumirá que os operadores selecionados foram bem-sucedidos. Ele continua com o restante do fluxo do processo.

#### **Sem êxito:**

Se uma condição de erro for atendida em tempo de execução, o CA Process Automation presumirá que os operadores selecionados falharam. Ele continua com o restante do fluxo do processo.

### **Desmarcado:**

Se uma condição de erro for atendida em tempo de execução, o CA Process Automation redefinirá os operadores selecionados e continuará com o fluxo do processo.

5. (Opcional) Para ignorar uma exceção e continuar o processo, adicione um operador Redefinir no modo de manipulador de exceções:

```
exceptionStart-<operator-name>.Source
```

**Observação:** quando você adiciona um campo à Lista de operadores Redefinir operador, os nomes de operador de loop do processo são exibidos na lista suspensa do novo campo. A opção Redefinir operador redefine todos os operadores no operador Loop e redefine esse operador para a primeira iteração. Após a redefinição, o operador Loop será reiniciado a partir da primeira iteração. Como o operador Loop não oferece suporte à simulação, o operador Redefinir *sempre* redefine um operador Loop. O operador Loop redefine e os seguintes valores de campo são ignorados:

- Continuar com resultado
- Condição de término
- Código de execução anterior
- Código de execução posterior

## Executar processos de forma interativa

É possível iniciar uma instância de um processo imediatamente ou em modo suspenso.

Quando um processo é iniciado imediatamente, uma instância do processo é criada, carregada na memória e inicia imediatamente os operadores de processamento. Se um processo for iniciado no modo suspenso, a instância do processo será carregada na memória, mas não iniciará o processamento.

Um atalho pode ser colocado para um processo em um objeto Exibição de processos. Se você for um usuário autorizado, poderá iniciar e monitorar o processo de dentro do objeto Exibição de processos. Na guia Operações da paleta Exibição de processos, clique com o botão direito do mouse no processo e clique em Iniciar.

É possível iniciar a versão atual de um processo, acessando o objeto de processo no Navegador da biblioteca. Um processo pode também ser iniciado enquanto você estiver editando-o no Criador de processos. Durante a edição de um processo, é possível disponibilizar as alterações e iniciar a versão atual do processo sem sair do Criador de processos.

Quando um processo é iniciado, o CA Process Automation cria uma cópia ou a *instância* dele na biblioteca de automação. As alterações para uma instância não afetam a definição básica do processo. É possível acessar a definição básica por meio do Navegador da biblioteca. As instâncias do processo são monitoradas por meio de uma Exibição de processos. Para monitorar uma instância do processo, é possível abrir a Exibição de processos na guia Operações da paleta Exibição de processos. É possível abrir o objeto de automação real da Exibição de processos por meio do Navegador da biblioteca.

**Mais informações:**

[A página Operações](#) (na página 421)

## Iniciar um processo da biblioteca

É possível iniciar um processo na guia Biblioteca. Iniciar um processo imediatamente permite que você execute uma tarefa em um ambiente de produção.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique em Orquestrador e selecione a combinação *Orquestrador:ambiente* apropriada.
3. Navegue até a pasta que contém o processo a ser iniciado.
4. Proceda de uma das seguintes maneiras:
  - Clique com o botão direito do mouse no processo e selecione Iniciar processo.
  - Selecione o processo e selecione Iniciar processo na lista suspensa Mais ações da barra de ferramentas.

O prompt Monitorar instância de processo é exibida.

5. Proceda de uma das seguintes maneiras:
  - Clique em Sim para abrir uma nova janela para monitorar a instância em execução do processo.
  - Clique em Não para executar uma instância do processo. O processo não é exibido.

O processo inicia imediatamente.

## Iniciar um processo da biblioteca como Suspenso

É possível iniciar uma instância de um processo em um estado suspenso para atingir um dos objetivos a seguir:

- Insira pontos de interrupção.
- Defina parâmetros.
- Faça outras alterações antes de o processo ser executado.
- Monitore ou controle a execução de um processo.
- Depure a sequência de etapas no processo.

### Siga estas etapas:

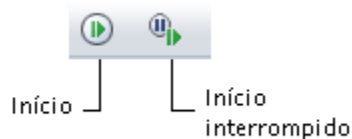
1. Clique na guia Biblioteca.
2. Clique em Orquestrador e selecione o *Orquestrador: ambiente* apropriado.
3. Navegue até a pasta que contém o processo a ser iniciado no estado suspenso.
4. Proceda de uma das seguintes maneiras:
  - Clique com o botão direito do mouse no processo e selecione Iniciar suspensos.
  - Selecione o processo e selecione Iniciar suspensos na lista suspensa Mais ações da barra de ferramentas.O prompt Monitorar instância de processo é exibida.
5. Proceda de uma das seguintes maneiras:
  - Clique em Sim para abrir a guia Criador com a barra de ferramentas de depuração. Você pode começar a trabalhar com a instância suspensa imediatamente.
  - Clique em Não para carregar o processo para a guia Operações. A guia Criador não é aberta. Posteriormente, você poderá navegar para a instância da paleta Exibição de processos na guia Operações. Para continuar, clique com o botão direito do mouse nessa instância e selecione Iniciar suspensos.

### Mais informações:

[Depurar um processo](#) (na página 439)

## Iniciar um processo durante a edição

Ao editar um objeto Processo, você poderá iniciar a versão atual sem sair do Criador de processos. Os comandos iniciar e Iniciar suspensos estão disponíveis no menu arquivo. Os botões Iniciar suspensos também estão disponíveis na barra de ferramentas do Criador de processos.



Os comandos Iniciar e Iniciar suspenso do Criador de processos criam uma instância do processo atual na memória, como iniciar um processo no Navegador da biblioteca. Se você clicar em Iniciar, a instância do processo correspondente será criada e iniciará a execução do processo imediatamente. Se você clicar em Iniciar suspenso, a instância do processo correspondente será criada, mas não iniciará a execução do processo.

Os comandos Iniciar e Iniciar suspenso solicitam o touchpoint no qual o processo será executado e, em seguida, perguntam se você deseja monitorar a execução do processo. Se você monitorar a execução do processo, o CA Process Automation abrirá uma janela separada do Criador de processos para trabalhar com a nova instância no modo de depuração.

## Abrir uma instância de um processo

O objeto Exibição de processos e a página Operações permitem visualizar instâncias de processos em um orquestrador. Você pode:

- Recupere e reinicie os processos que foram suspensos após um incidente.
- Avalie os valores das variáveis de conjunto de dados e o status dos operadores nos processos em execução, suspensos ou encerrados.

É possível criar objetos Exibição de processos ou usar a página Operações para monitorar e editar instâncias de processos. Clique na guia Operações para monitorar todas as instâncias de processos em um orquestrador. Você também pode definir filtros para monitorar apenas os objetos selecionados.

### Mais informações:

[A página Operações](#) (na página 421)

## Estados do processo

O Criador de processos atualiza periodicamente o estado atual do processo. Os ícones codificados por cor são usados para indicar o estado de cada operador. É possível editar o processo enquanto ele estiver em execução ou suspenso. Depois que um processo for concluído ou cancelado, você não poderá mais alterar a instância.

## Depurar um processo

Quando você abre uma instância de um processo, o Criador de processos o ajuda a monitorar o status e a depurar um processo. Os botões de depuração estão disponíveis na barra de ferramentas do Criador de processos. Quando o processamento for suspenso, você poderá editar o processo e alterar os valores de parâmetro nos operadores.

Um processo pode ser suspenso de várias maneiras:

- Quando uma instância é iniciada em um estado suspenso.
- Quando você clica no botão suspender processo.
- Quando um processo termina e você clica no botão Alterar o status do processo ao concluir, na barra de ferramentas do Criador de processos. O processo está suspenso, mas o status é exibido como Bloqueado.
- Quando não há quaisquer operadores válidos em qualquer ramificação de um processo em execução. O processo está suspenso, mas o status é exibido como *Bloqueado*.

## Suspender um processo

Quando um processo estiver no estado suspenso, você pode executar um dos procedimentos a seguir:

- Alterar se um processo será descarregado após a conclusão
- Redefinir o processo
- Redefinir os operadores em um processo
- Adicionar ou remover pontos de interrupção
- Modificar o processo
- Clique no botão **Retoma processo** para continuar o processamento
- Cancelar o processo

A modificação de um processo em um estado suspenso permite que você trabalhe em um problema imprevisto e, em seguida, retome a execução automatizada enquanto ainda estiver rastreando todas as alterações feitas. Alterne para uma Exibição de processos para exportar uma instância modificada de um processo e salvar permanentemente quaisquer alterações feitas em tempo de execução.

Para suspender a execução de uma instância de um processo ao trabalhar no modo de depuração, clique no botão **Suspende processo** na barra de ferramentas do Criador de processos.

A execução da instância do processo é interrompida. Nenhuma outra dependência será examinada até que a execução do processo seja retomada. É possível editar um processo em um estado suspenso. Todas as modificações nos parâmetros do operador ou em outros elementos do processo afetarão apenas essa instância do processo.

Para reiniciar a execução em uma instância suspensa de um processo, clique no botão **Retoma processo**. O botão **Retoma processo** reiniciará um processo suspenso a partir de onde a execução foi interrompida, a menos que ele seja redefinido. Se uma instância de um processo suspenso for redefinida, a execução será reiniciado desde o início do processo.

## Alterar se processos são descarregados na conclusão

Durante a execução de um processo no modo de depuração, o módulo de fluxo de trabalho normalmente não descarrega a instância do processo quando ele atinge um operador Interromper. Isso lhe permite modificar e reiniciar o processo.

Para forçar o módulo de fluxo de trabalho a descarregar um processo quando ele atingir um operador Interromper, clique em Manter estado no menu Controle para desmarcar a caixa de seleção ao lado do comando. Também é possível usar o botão Alterar o status do processo ao concluir, na barra de ferramentas para ativar ou desativar esse comando. O botão da barra de ferramentas permanece inativo enquanto o comando está ativado.

## Definir e remover pontos de interrupção de um processo

É possível usar os pontos de interrupção para identificar os erros. Os pontos de interrupção o ajudam a verificar os valores das variáveis e os parâmetros do operador. Defina um ponto de interrupção em um operador para interromper um processo imediatamente antes de o operador ser iniciado. É possível definir os valores de parâmetro e examinar o processamento de um operador à medida que ele ocorrer.

Quando um ponto de interrupção é definido, todo o processo é suspenso quando atinge o operador com o ponto de interrupção. Um símbolo de ponto de exclamação (!) é exibido ao lado do operador que suspendeu o processo.

É possível definir e remover pontos de interrupção em um objeto de processo ou em uma instância de um processo suspenso. Os pontos de interrupção definidos na definição do objeto de processo original são exibidos automaticamente em todas as instâncias do processo.

**Siga estas etapas:**

1. Abra um processo no Criador de processos.
2. Selecione um ou mais operadores no processo.
3. Na barra de ferramentas, clique em Definir pontos de interrupção.  
O símbolo de ponto de interrupção é exibido ao lado do operador selecionado.
4. Para remover pontos de interrupção existentes, selecione um ou mais operadores e clique no botão Remover pontos de interrupção na barra de ferramentas de Depuração.

## Depurar um processo Java

O conector Java usa o *Apache Log4j* para capturar as mensagens de log do conector. Ao solucionar um problema com um processo Java, depure-o, ativando e, em seguida, revendo os arquivos de log. As mensagens do log capturadas no nível DEBUG são muito detalhadas e devem ajudar os engenheiros de sistema a definir a causa raiz de um problema.

**Observação:** os caminhos dos arquivos log4j.xml e c2o.log são alterados durante a execução do módulo Java em um agente do CA Process Automation.

### Siga estas etapas:

1. Localize o arquivo log4j.xml neste caminho:  
Caminho\_de\_instalação\_do\_CA Process Automation\Domain\server\c2o\conf\log4j.xml

2. Defina o nível do limite de log4j do módulo Java como DEBUG.

Um exemplo é apresentado a seguir e mostra a seção e a linha específica (em negrito) do arquivo log4j.xml:

```
<!-- A size based file rolling appender for C20 and JXTA Logs-->
  <appender name="C20FILE"
class="org.jboss.logging.appender.RollingFileAppender">
  <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
  <param name="File" value="{jboss.server.home.dir}/log/c2o.log"/>
  <param name="Threshold" value="DEBUG"/>
  <param name="Append" value="true"/>
  <param name="MaxFileSize" value="50000KB"/>
  <param name="MaxBackupIndex" value="3"/>
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d %-5p [%c] [%15.15t] %m%n"/>
  </layout>
</appender>
```

3. Insira a seção a seguir no arquivo log4j.xml:  
<category name="com.optinuity.c2o.servicegroup.javaobject">  
 <priority value="DEBUG" />  
</category>
4. Abra as mensagens do conector Java capturadas no arquivo c2o.log localizado neste caminho:  
Caminho\_de\_instalação\_do\_CA Process Automation\Domain\server\c2o\log\c2o.log

## Redefinir um processo

É possível redefinir uma instância suspensa de um processo para reiniciar a execução. Todas as variáveis, todos os parâmetros e todos os operadores em uma instância suspensa de um processo são redefinidos para seus estados iniciais, com uma exceção: o aplicativo não redefine os parâmetros definidos pelo usuário.

Se você continuar a execução de um processo após redefini-lo, o processamento reiniciará no operador Iniciar. É possível redefinir um processo durante os testes ou a depuração para executar novamente as cadeias de operadores. Na produção, é possível redefinir um processo após um incidente. É possível modificar um processo de redefinição antes de reiniciá-lo para evitar que algumas tarefas sejam refeitas ou para executar algumas tarefas adicionais. Por exemplo, você pode:

- Definir ou remover pontos de interrupção
- Usar um processo diferente
- Definir configurações de parâmetro do operador

## Cancelar um processo

Para interromper uma instância de um processo, clique no botão Cancelar processos na barra de ferramentas do Criador de processos.

**Observação:** não será possível modificar uma instância de processo após a execução do comando cancelar.

## Controlar a ramificação de um processo

Os comandos a seguir permitem que você controle ramificações individuais de um processo sem afetar todo o processo.

- ativar e desativar operadores
- cancelar a execução dos operadores
- redefinir os operadores individuais e retomar a execução dos operadores em um processo.

Esses comandos são exibidos na barra de ferramentas do Criador de processos para instâncias e criadores de processos.

## Desativar operadores ou desativar ramificações

É possível usar o comando Desativar operadores para desativar um operador e desativar temporariamente parte de um processo sem modificar absolutamente o processo. A desativação de um operador interrompe sua execução e de todos os operadores subsequentes em uma ramificação de um processo. Um operador fica inacessível a um processo quando cada caminho que leva a ele de qualquer operador Iniciar passa por um operador desativado. O comando Desativar operadores estará disponível durante a edição do objeto de definição para um processo ou uma instância de um processo.

### Para desativar um operador em um processo

1. Abra um objeto de processo ou uma instância de um processo.

**Observação:** para uma instância de um processo em execução, suspenda a execução antes de prosseguir para a próxima etapa do procedimento.

2. Selecione um ou mais operadores que deseja desativar.
3. Clique em Desativar operadores na barra de ferramentas do Criador de processos para desativar um operador.

O símbolo de desativado é exibido ao lado do operador.

**Observação:** para ativar operadores desativados, selecione um ou mais operadores desativados e clique em Ativar operadores na barra de ferramentas.

Você também pode desativar uma ramificação inteira clicando com o botão direito do mouse em um link e selecionando Desativar link no menu.

### Mais informações:

[Simular o processamento dos operadores](#) (na página 446)

## Cancelar um operador

O comando Cancelar operadores cancela o processamento de um ou mais operadores selecionados que estiverem em um estado *Em execução*. Um operador cancelado passa para o estado *Com falha*. O processamento não continuará em uma ramificação cancelada de um processo, mas você ainda poderá continuar em outras ramificações. Se você tiver definido um manipulador de exceções, ele será executado para lidar com uma exceção de cancelamento.

**Siga estas etapas:**

1. Abra uma instância em execução de um processo.
2. Selecione o operador em execução a ser cancelado.
3. Clique no botão cancelar operadores na barra de ferramentas do designer de processos:  
O processamento é cancelado para o operador.

## Redefinir os operadores em um processo

O comando Redefinir operadores redefine os operadores selecionados em um processo suspenso para seus estados iniciais como se eles não tivessem sido executados. Esse comando é geralmente usado para permitir que os operadores em uma instância de um processo sejam executados novamente sem precisar redefinir todo o processo. Você pode modificar os operadores com diferentes configurações de parâmetro ou modificar o processo de outras maneiras.

**Siga estas etapas:**

1. Abra uma instância suspensa de um processo.  
Se necessário, suspenda a execução de uma instância em execução.
2. Selecione um ou mais operadores que deseja redefinir.
3. Clique no botão Redefinir operadores na barra de ferramentas do Criador de processos.  
Os operadores são redefinidos.  
Qualquer operadorLoop será redefinido da seguinte forma:
  - Redefine todos os operadores dentro do operador Loop.
  - Redefine o operador Loop para a primeira iteração.Após a redefinição, o operador Loop será reiniciado a partir da primeira iteração.

## Reiniciar a execução de um processo suspenso

O comando Retoma processo continua o processamento dos operadores redefinidos em uma instância suspensa de um processo.

**Para retomar a execução de um processo suspenso**

1. Abra uma instância suspensa de um processo.
2. Clique no botão reiniciar processo na barra de ferramentas do designer de processos.  
O processo é retomado.

## Simular o processamento dos operadores

Para ignorar um operador ao executar os operadores subsequentes em um processo, você poderá *simular* a execução do operador, em vez de desativá-lo. É possível simular o processamento dos operadores selecionados ou de um processo inteiro. Você pode usar as propriedades de simulação de processo para qualquer operador em um processo. Você também pode substituir as configurações de processo para operadores de processo individuais.

O modo de simulação é geralmente usado durante o desenvolvimento para verificar o fluxo dentro de um processo ou a sincronização de processos interdependentes. No ambiente de produção, o modo local é geralmente usado para ignorar tarefas individuais, de modo a acomodar as condições transitórias no sistema, como um agente indisponível, um orquestrador ou um aplicativo suspenso. O modo de simulação distante é usado principalmente na fase de desenvolvimento para verificar a configuração e os parâmetros dos operadores em um processo. O modo de simulação distante também pode ser usado para executar tarefas simuladas em produção periodicamente, para a exclusiva finalidade de verificar os valores de campo do conjunto de dados.

O modo de simulação controla os resultados e a duração de um operador em um processo. É possível usá-lo durante o desenvolvimento para evitar a execução de um operador durante o teste dos operadores nas ramificações subsequentes de um processo. A simulação é útil em um ambiente de produção quando um operador não tiver que ser executado. Os operadores não são executados quando os processos ou os recursos estão indisponíveis. Em qualquer caso, a simulação de um operador permitirá executar um processo sem alterar os links de entrada ou saída no operador. Em outras palavras, você conseguirá evitar a execução do operador temporariamente preservando, ao mesmo tempo, a estrutura e as dependências no seu processo.

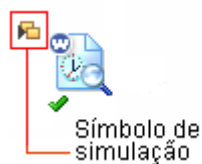
## Simular o processamento de um operador selecionado

É possível simular o processamento de um operador selecionado.

### Siga estas etapas:

1. Abra qualquer operador no Navegador da biblioteca ou um processo aberto.
2. Configure as propriedades da simulação para qualquer operador.
3. Se a simulação não estiver ativada para o processo, marque primeiro a caixa de seleção *Substituir opções de simulação no processo*.
4. Para ativar a simulação, especifique um tipo de simulação Local ou Distante.
  - Local: o operador não é processado. O CA Process Automation não chama o módulo associado ou verifica os parâmetros do módulo. O operador simulado retorna o resultado e processa o link especificado pela opção Condição de término.
  - Distante: o módulo de fluxo de trabalho chama o módulo associado. O módulo verifica os parâmetros antes de retornar o resultado, mas não executa realmente o operador. Se os parâmetros estiverem incorretos, o operador simulado falhará, independentemente do resultado especificado. Se os parâmetros estiverem corretos, o operador simulado retornará o resultado e processará o link especificado pela opção Condição de término.

O símbolo de simulação aparece ao lado de um operador quando a simulação local ou distante é ativada:



5. (Opcional) Marque a caixa de seleção *Avaliar código de execução anterior e posterior* para evitar os efeitos acarretados quando tais códigos são ignorados. Selecione essa opção para avaliar as precondições do operador e executar seu código de execução posterior.
6. Defina o parâmetro de condição de término. Ele especifica o resultado simulado de um operador. A condição de término determina qual link de saída é processado após o processamento simulado do operador. A definição desse parâmetro determina o processamento subsequente das ramificações dependentes no processo. Essa configuração determina o valor retornado pela variável *Resultado* de um operador. A configuração *Com êxito* ou *Concluído* retorna um valor positivo, que ativa um link de saída Com êxito em um operador. A configuração *Falha* retorna zero ou um valor negativo, que ativa um link de saída Falha ou Cancelar. A configuração *Resultado personalizado* permite retornar um valor arbitrário, normalmente usado para testar um link de saída personalizado em um operador.

7. (Opcional) Especifique um atraso em segundos para simular a duração da tarefa. Altere o campo Atrasar do valor padrão zero se o tempo normalmente necessário para executar um operador puder afetar o comportamento ou o resultado de outros operadores no processo.

**Mais informações:**

[Código de execução anterior e código de execução posterior](#) (na página 234)

## Simular o processamento de um processo inteiro

É possível simular o processamento de todas as etapas de um processo. As configurações da simulação do processo se aplicam a todos os operadores no processo que não estão configurados para substituir as configurações de simulação do processo.

**Siga estas etapas:**

1. Abra o processo.
2. No menu Arquivo, clique em Propriedades ou clique no botão Propriedades da barra de ferramentas do Criador de processos:



3. Na guia Simulação do painel Propriedades, defina as configurações de simulação padrão para todos os operadores no processo:
  - Tipo
  - Atrasar
  - Condição de término
  - Avaliar as pré-condições e o código de pós-execução

Exceto para a opção que substitui as configurações de processo, essas configurações são as mesmas descritas para configurar um operador.

Os parâmetros da simulação de um processo são os mesmos de um operador, exceto pelo fato de que são as configurações padrão para cada operador em um processo. Para alterar as configurações de propriedade de simulação de um operador específico, marque a caixa de seleção Substituir opções de simulação no processo nas páginas de propriedades do operador.

# Capítulo 11: Gerando e usando a documentação do processo

---

É possível gerar a documentação dos processos automatizados. Outros poderão imprimir a documentação do processo que você gerar.

Esta seção contém os seguintes tópicos:

[Sobre a documentação do processo](#) (na página 449)

[Gerar documentação do processo](#) (na página 450)

## Sobre a documentação do processo

Você pode gerar detalhes da definição do processo e gráficos com anotações. O formato da saída é PDF.

A documentação do processo inclui detalhes nos seguintes níveis:

- Detalhes em nível de processo
- Detalhes em nível de operador para cada operador usado no fluxo do processo

A lista a seguir resume os detalhes que são gerados em cada nível:

### **Detalhes em nível de processo**

A documentação do processo gera os seguintes detalhes para um processo selecionado:

- Imagem do processo
- Nome do processo
- Parâmetros criados pelo usuário no conjunto de dados do processo
- Caminho da biblioteca
- Proprietário
- Versão do processo
- Release
- Reservado por
- Data de criação
- Data da última modificação
- Orquestrador em que o processo reside
- Qualificadores do processo

### Detalhes de nível do operador

A documentação do processo gera os seguintes detalhes para cada operador que um processo selecionado usa:

- Bitmap do ícone que representa o operador
- Os parâmetros criados pelo usuário no conjunto de dados do operador
- Nome do operador
- Descrição do operador
- O tipo de operador (isto é, a categoria do operador)
- Parâmetros do operador
- Portas de saída
- Configurações de execução
  - Código de execução anterior
  - Código de execução posterior
  - Loop
  - Tempo de expiração
  - Destino

## Gerar documentação do processo

É possível gerar a documentação do processo a qualquer momento. Por exemplo, use esse recurso para registrar o status de um processo antes de fazer sua transição para produção.

### Siga estas etapas:

1. Abra o processo na guia Biblioteca.
2. Na guia Criador, clique no botão Processar documentação da barra de ferramentas.



A caixa de diálogo Opções da documentação do processo é aberta.

3. (Opcional) Selecione opções para incluir na documentação.
  - Conjunto de dados do processo
  - Informações do operador
  - Código Pre-/Post-Execution
  - Propriedades do operador
  - Conjunto de dados do operador
  - Manipulador de exceções (se for selecionado, o CA Process Automation gera documentação e um arquivo de imagem para todos os operadores no Manipulador de exceções)
  - Manipulador de mudança de rota (se for selecionado, o CA Process Automation gera documentação e um arquivo de imagem para todos os operadores no Manipulador de mudança de rota)
4. Clique em OK.

Uma caixa de diálogo do navegador perguntará se deseja abrir ou salvar o arquivo.
5. Clique em Abrir para exibir o conteúdo do arquivo zip, ou clique em Salvar para salvar o arquivo zip em um local de sua escolha.

O arquivo zip contém:

  - Um arquivo PDF da documentação que contém o processo.
  - As imagens no tamanho normal contidas na documentação do processo. O PDF contém imagens em escala, portanto, eles poderão não ser legíveis se o processo for muito grande.
  - Um arquivo XML que contenha os dados brutos usados para gerar a documentação no formato XML. Esse arquivo permite transformar XML em outros formatos.



# Capítulo 12: Liberar objetos para outro ambiente

---

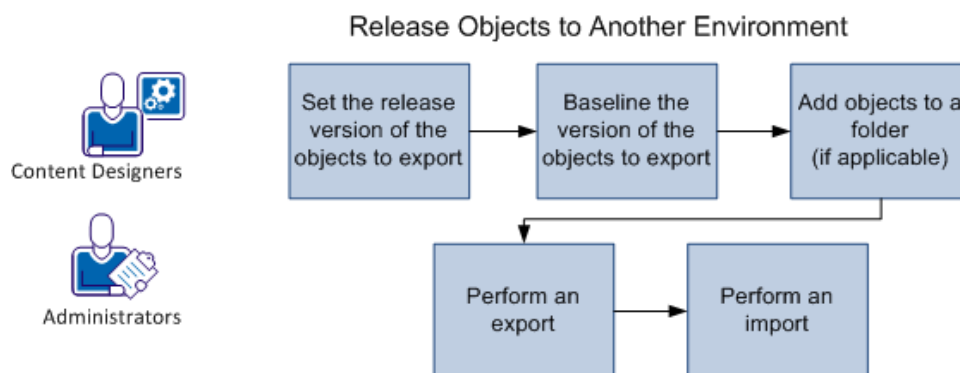
Depois de criar e testar um novo processo, você pode liberá-lo com os objetos que ele utiliza. Normalmente, você libera um novo processo a partir de uma biblioteca do orquestrador no ambiente de design para uma biblioteca do orquestrador no ambiente de produção. Você pode exportar objetos [sozinhos, em uma pasta, em um pacote ou em um pacote de conteúdo](#) (na página 460). No entanto, para exportar objetos relacionados em um único recipiente e anexar uma versão da release a eles, você deve exportar esses objetos como pacote de conteúdo.

Se você usou pacotes no CA Process Automation 04.1.00, migre esses objetos para um novo pacote de conteúdo.

O ambiente de criação e o ambiente de produção estão no mesmo domínio. Se a funcionalidade for criada em um local para ser implantada em um local diferente, você pode implementar o produto em dois domínios separados. É possível importar o arquivo XML para um objeto, uma pasta, um pacote ou um pacote de conteúdo em um orquestrador em outro ambiente em um domínio diferente.

**Observação:** quando você exportar objetos para importação em um orquestrador no mesmo ambiente, poderá omitir os objetos definidos como versão base e definir versões de release.

Um administrador ou criador de conteúdo pode executar esse processo. Normalmente, um administrador exporta uma pasta como pacote de conteúdo do ambiente padrão e, em seguida, o importa para um ambiente de produção. A ilustração a seguir descreve o processo:



**Siga estas etapas:**

1. [Definir a versão de release dos objetos a serem exportados.](#) (na página 455)
2. [Definir a versão base de objetos a serem exportados \(se aplicável\)](#) (na página 457).
3. [Adicionar objetos a uma pasta \(se aplicável\)](#) (na página 465).
4. [Executar uma exportação](#) (na página 461).
5. [Executar uma importação](#) (na página 477).

## Versões da release

Uma *release* ocorre quando você implanta um processo e seus componentes a partir de um ambiente de criação para um ambiente de produção. Ao selecionar os componentes para compactar para exportação, selecione uma versão de cada componente. É possível inserir um valor específico da release no campo Versão da release da versão de cada pasta e objeto que você planeja empacotar para exportação. Essa prática é necessária quando você exportar objetos para uma pasta como pacote de conteúdo.

Depois de definir uma versão da release para cada objeto, é possível adicionar uma versão correspondente à pasta que deseja exportar como pacote de conteúdo.

Pacotes de conteúdo começam como pastas no ambiente inicial do CA Process Automation. Ao exportar uma pasta como pacote de conteúdo e importá-la para um novo ambiente, o produto define a versão base do objeto importado no pacote de conteúdo. Os usuários não podem modificar os dados da Versão da release de objetos importados.

## Exibir informações da versão de release

A guia Versões exibe informações sobre as versões de release definidas para o objeto selecionado.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione um objeto na janela principal e clique em Propriedades.
3. O painel Propriedades é exibido na parte inferior da janela.
4. Clique na guia Versão.
5. Verifique os dados exibidos em Versão da release.

A Versão da release identifica a versão específica de um objeto que foi importado ou que você deseja exportar e, em seguida, importar para um ambiente de produção. Você pode definir esse identificador em qualquer objeto individual.

O produto bloqueia esse campo nas seguintes circunstâncias:

- Quando o objeto tiver sido importado como parte de um pacote de conteúdo.
- Quando o objeto tiver sido importado de um pacote (na release 4.1.00) com sua versão da release no modo não modificável.

**Observação:** consulte a *Referência de interface de usuário* para obter informações detalhadas sobre a guia Versões.

## Definir a versão da release de objetos a serem exportados

A guia Versões exibe informações sobre as versões de release definidas para o objeto selecionado. É possível definir a versão da release quando estiver pronto para implantar um objeto em um ambiente de produção. Uma versão da release de um objeto geralmente identifica a versão específica para a implantação. Por meio do campo Versão da release, por exemplo, você poderá definir o atributo de versão da release para um processo e cada objeto associado.

Quando você estiver pronto para implantar um objeto em um ambiente de produção:

- Você poderá exportar um único objeto.
- É possível exportar todos os objetos em uma pasta selecionada, incluindo as subpastas.
- Você poderá exportar os objetos como pacote de conteúdo. Nesse caso, as versões de release serão necessárias em todas as pastas que você exportar como pacote de conteúdo e em todos os objetos incluídos na pasta. Quando o pacote de conteúdo for importado, a versão da release não poderá ser modificada no novo ambiente, a menos que seja feita uma cópia do objeto.

**Observação:** consulte o tópico [Qual é a diferença entre a exportação de um objeto, uma pasta, um pacote de conteúdo e um pacote](#) (na página 460) para obter mais informações.

Em cada caso, o arquivo exportado contém o valor do atributo da versão da release de cada objeto que está sendo exportado.

É possível definir a versão da release das seguintes maneiras:

#### **A partir da biblioteca**

##### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Selecione uma pasta no painel esquerdo e, em seguida, um único objeto da lista no painel direito.
3. Clique com o botão direito do mouse no objeto e, em seguida, selecione Propriedades.
4. No painel Propriedades, clique na guia Versões.

**Observação:** consulte as dicas de ferramenta ou a *Referência de interface de usuário* para obter descrições dos campos e exemplos.

5. Clique com o botão direito do mouse na versão aplicável e, em seguida, selecione Definir a versão da release.
6. Digite um valor adequado no campo Versão da release.

#### **A partir da guia Versões no Editor de objetos**

##### **Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique duas vezes em um objeto.

A versão atual do objeto é exibida no editor de objetos.

3. Clique na guia Versões.

**Observação:** o editor de processos não contém uma guia Versões.

4. Clique com o botão direito do mouse na versão do objeto para implantar e selecione Definir a versão da release.

O campo Versão da release será exibido.

5. Digite uma versão da release para o objeto e, em seguida, pressione Enter.

#### **Ao disponibilizar um objeto**

Consulte [Disponibilizar um objeto](#) (na página 87).

## Definir a versão base da versão da release dos objetos

Um criador de conteúdo define a versão base de um objeto para criar um instantâneo de versão no ambiente de criação. Os usuários não podem alterar a versão base de um objeto, mas podem usá-la como base para uma nova versão. Um criador de conteúdo pode definir a versão base de objetos a qualquer momento, mas, em geral, define a versão base do objeto destinado à exportação. Na preparação para exportar, os criadores de conteúdo podem definir a versão base e definir a versão da release na versão de exportação na mesma sessão.

**Observação:** a versão de um objeto para o qual você define a versão base para exportação em uma pasta como pacote de conteúdo pode ser a atual ou outra versão. Em geral, é a versão atual que é marcada.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique em Orquestrador e selecione o *Orquestrador:ambiente* apropriado.
3. Vá até a pasta com um objeto de destino e verifique se o campo Estado do objeto indica Disponibilizado. (Em caso negativo, disponibilize o objeto antes de continuar.)
4. Selecione o objeto de destino e clique em Propriedades.
5. Clique na guia Versões.
6. Se o botão Versão base estiver ativado na versão a ser liberada, clique em Versão base.

No exemplo a seguir, o botão Versão base para Rv2.0 está ativado.

*Equation 3: As propriedades de todos os objetos possuem uma guia Versões. Nela, é possível definir uma versão da release, definir uma versão como atual e definir a versão base de uma ou mais versões listadas.*

Propriedades				
<span>Geral</span> <span>Tags</span> <span>Versões</span> <span>Trilha de auditoria</span>				
	Versão	Versão da release	Atual	Versão base
+	2	Rv2.0	Atual	Versão base
+	1	Rv1.0	Atual	Versão base

7. Feche o painel Propriedades.

## Definir a versão da release do pacote de conteúdo como atual

A guia Versões exibe informações sobre as versões de release que estão disponíveis para um pacote de conteúdo. A versão da release normalmente identifica uma determinada versão do pacote para implantar em um ambiente de produção.

Após importar o pacote de conteúdo no novo ambiente, é possível usar a guia Versões para definir o atributo de versão da release do pacote de conteúdo.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse em um pacote de conteúdo e, em seguida, selecione Propriedades.
3. No painel Propriedades, clique na guia Versões.
4. Para definir o pacote de conteúdo como a versão da release atual, selecione-o e clique em Atual.
5. Clique em Sim na mensagem de aviso.

O navegador da biblioteca exibe a versão atual do pacote de conteúdo.

## Exportando e importando

Para mover os seguintes itens da biblioteca de um orquestrador para a biblioteca de outro orquestrador, use os processos de exportação e importação:

- Objetos
- Pastas
- Pacotes
- Pacotes de conteúdo

Por exemplo, você pode copiar de um ambiente de desenvolvimento para um ambiente de produção ou de um orquestrador para outro orquestrador em um mesmo ambiente. O processo de exportação cria um arquivo XML utilizado pelo processo de importação. Depois de exportar os objetos, você deve importá-los no orquestrador de destino.

Considere os seguintes pontos antes de exportar e importar:

- O arquivo de exportação preserva informações completas, incluindo hierarquias e caminhos, para cada objeto exportado. Durante a exportação, você pode salvar toda a hierarquia absoluta de pastas a partir da pasta raiz ou apenas os caminhos relativos que começam na pasta selecionada.
- Quando você importa um arquivo XML que foi exportado como caminho Absoluto ou Relativo para uma biblioteca de automação, o CA Process Automation reconstrói as informações de caminho (absoluto ou relativo) no arquivo a partir da estrutura da pasta exportada.
- Quando você exportar a pasta raiz, o produto exportará toda a biblioteca de automação. Verifique as restrições de tamanho antes de exportar a pasta raiz.
- É possível abrir um arquivo de exportação em qualquer editor de texto XML.
- Não é possível modificar a versão da release de um pacote de conteúdo exportado. O processo de exportação criptografa o arquivo de exportação XML para impedir modificações.
- Ao exportar uma pasta ou um objeto fora de um pacote de conteúdo, o próprio objeto determinará o estado do objeto importado. Por exemplo, se você exportar um processo não modificável fora de um pacote de conteúdo, o processo continuará não modificável após a importação.

## Exportando

Os tópicos a seguir explicam o processo de exportação.

## Qual é a diferença entre a exportação de um objeto, uma pasta, um pacote de conteúdo e um pacote?

O CA Process Automation permite exportar objetos e pastas de várias maneiras. Cada opção de exportação cria um arquivo XML que pode ser importado para um ambiente diferente.

É possível exportar os seguintes itens:

### **Um objeto sozinho**

Só é possível exportar um objeto no modo modificável. Você pode adicionar uma versão da release ao objeto, mas não é obrigatório. Depois de importar um objeto para um novo ambiente, é possível modificar o objeto diretamente.

### **Uma pasta e todas as subpastas e objetos**

A exportação da pasta exporta a pasta e seu conteúdo, incluindo todas as subpastas. O conteúdo pode incluir qualquer objeto disponível na biblioteca do CA Process Automation.

É possível exportar uma pasta no modo modificável para que você possa modificar a pasta e seu conteúdo diretamente em um novo ambiente. O produto não requer que os objetos que foram exportados por meio de uma pasta tenham uma versão da release.

O produto preserva o caminho para a pasta e a estrutura hierárquica dos objetos e pastas secundárias no arquivo de exportação.

### Uma pasta como pacote de conteúdo

Uma pasta torna-se um pacote de conteúdo após a exportação e não poderá ser modificada no novo ambiente. O produto preserva o caminho para a pasta e a estrutura hierárquica dos objetos e pastas secundárias no arquivo de exportação.

Você pode adicionar informações da release a uma pasta que deseja exportar como pacote de conteúdo. O atributo de versão da release é necessário para todas as pastas, mas também é possível adicionar outras informações relevantes sobre a pasta ou seus objetos.

A prática recomendada é exportar uma pasta como pacote de conteúdo quando você estiver realizando a transição de um novo processo com seus componentes do ambiente de desenvolvimento para o ambiente de produção. Os destinatários no ambiente de produção não podem modificar o conteúdo do pacote de conteúdo.

**Importante:** se você não definir a versão da release do pacote de conteúdo ou dos objetos filho na pasta, o produto cancelará a tarefa de exportação.

### Um pacote

O objeto do pacote é suportado somente para compatibilidade com versões anteriores no CA Process Automation 04.2.00. Você não pode fazer referência a pacotes em formulários de solicitação de interação, formulários de solicitação inicial, operadores personalizados, exibições de processos ou conjuntos de dados. Você ainda pode exportar um pacote da 04.1.00, mas não pode modificar o pacote depois de importá-lo para um novo ambiente. No entanto, é possível modificar os objetos do pacote.

Para agrupar objetos similares e exportá-los para um novo ambiente, use um pacote de conteúdo.

Pacotes que você migrar da release 04.1.00 para a 04.2.00 serão exportados da mesma maneira que um objeto.

**Observação:** não é possível exportar várias pastas ou objetos de automação simultaneamente. Em vez disso, crie uma pasta e, em seguida, reúna todos os objetos que serão exportados como um único pacote de conteúdo. Não é possível exportar como pacote de conteúdo uma pasta que já contém um pacote de conteúdo.

## Executar uma exportação

Para mover objetos, pastas, pacotes e pacotes de conteúdo da biblioteca de um orquestrador para a biblioteca de outro orquestrador, use o processo de exportação. O orquestrador pode estar no mesmo ambiente ou em um ambiente diferente.

## Exportar um objeto

**Observação:** pacotes que você migrar da release 04.1.00 para a 04.2.00 serão exportados da mesma maneira que um objeto.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique em Orquestrador e, em seguida, selecione o *Orquestrador:ambiente* apropriado.
3. Vá até o objeto que deseja exportar, clique com o botão direito do mouse no objeto e, em seguida, selecione Exportar.
4. Para especificar os caminhos, selecione uma das seguintes opções:

**Exportar, Caminhos absolutos**

Inclui o caminho completo do objeto selecionado.

**Exportar, Caminhos relativos**

Inclui o caminho relativo para a pasta que contém o objeto selecionado.

5. Para salvar o arquivo XML, clique em Salvar na caixa de diálogo Download de arquivo.

O nome de arquivo padrão é nome-do-objeto.xml.

6. Vá até o local em que deseja salvar o arquivo XML em sua unidade local.
7. Defina o nome com o qual deseja salvar o arquivo.

Por exemplo, anexe *\_RP* ao nome do arquivo para indicar um caminho relativo ou *\_AP* para indicar um caminho absoluto.

*nome-do-objeto\_RP.xml*

*nome-do-objeto\_AP.xml*

8. Clique em Salvar.  
O produto exporta o objeto.

## Exportar uma pasta

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique em Orquestrador e, em seguida, selecione o *Orquestrador:ambiente* apropriado.

- Vá até a pasta ou o objeto a ser exportado, clique com o botão direito do mouse na pasta e, em seguida, selecione uma das seguintes opções:

**Exportar**

Exporta uma pasta ou um pacote no modo modificável.

**Exportar como pacote de conteúdo (somente pastas)**

Exporta uma pasta como pacote de conteúdo no modo não modificável.

- Para especificar os caminhos de exportação, selecione uma das seguintes opções:

**Exportar, Caminhos absolutos**

Inclui o caminho completo da pasta selecionada.

**Exportar, Caminhos relativos**

Inclui o caminho relativo para a pasta que contém a pasta selecionada.

**Importante:** se a versão da release do pacote de conteúdo ou de seus objetos filho não estiver definida na pasta, o produto cancelará a tarefa de exportação.

- Para salvar o arquivo XML, clique em Salvar na caixa de diálogo Download de arquivo.

O nome de arquivo padrão é nome-da-pasta.xml.

- Vá até o local em que deseja salvar o arquivo XML em sua unidade local.

Defina o nome com o qual deseja salvar o arquivo. Por exemplo, anexe `_RP` ao nome do arquivo para indicar um caminho relativo ou `_AP` para indicar um caminho absoluto.

`folder-name_RP.xml`

`folder-name_AP.xml`

- Clique em Salvar.

O CA Process Automation exporta a pasta e seu conteúdo.

**Mais informações**

[Exportar caminhos relativos ou Exportar caminhos absolutos](#) (na página 468)

## Objetos de pacote de conteúdo

Os processos complexos podem utilizar vários objetos, incluindo operadores, ícones, recursos, conjuntos de dados e até mesmo outros processos. Eles podem ser distribuídos entre várias pastas com objetos não relacionados em uma biblioteca de automação. Esta complexidade apresenta os seguintes possíveis problemas:

- Um usuário poderá ter dificuldade para trabalhar apenas com os objetos necessários para uma implantação ou atualização posterior.
- Um desenvolvedor que esteja trabalhando em diferentes pastas com diferentes objetos poderá ter dificuldade para rastrear ou localizar objetos específicos.
- Um administrador que exporte objetos atualizados para implantação em um ambiente de produção poderá esquecer facilmente de um objeto necessário.

Para exportar todos os objetos de automação relacionados, você pode reuni-los em uma pasta e exportar essa pasta como pacote de conteúdo para outro ambiente do CA Process Automation. Pacotes de conteúdo são úteis porque permitem agrupar objetos relacionados em um único recipiente e, em seguida, atribuir uma versão da release a eles. Em seguida, quando os objetos forem importados para um novo ambiente de produção, eles não poderão ser modificados, a menos que os objetos importados sejam copiados. Não é possível editar ou excluir nenhum dos objetos no pacote de conteúdo. Isso impede que os objetos sejam alterados de forma potencialmente perigosa. Em outras palavras, sempre é possível reverter para o pacote de conteúdo original se desejar reverter alterações indesejadas feitas nos objetos. Como as versões de release dos objetos não são modificáveis, os criadores de conteúdo no ambiente de criação de origem podem identificar as versões de objetos exportados e importados se for necessário solucionar problemas.

Como comparação, se você exportar objetos sozinhos ou por meio de uma pasta, eles poderão ser modificados em um novo ambiente. Os objetos que são importados sozinhos ou por meio de uma pasta (em vez de um pacote de conteúdo) não precisam ter uma versão da release. Esses objetos podem ser modificados diretamente. A prática recomendada é exportar uma pasta como pacote de conteúdo quando você estiver realizando a transição de um novo processo com seus componentes do ambiente de desenvolvimento para o ambiente de produção.

**Observação:** não é possível exportar várias pastas ou objetos de automação de uma vez. Crie uma pasta e, em seguida, reúna todos os objetos que serão exportados como um único pacote de conteúdo. Não é possível exportar como pacote de conteúdo uma pasta que já contiver um pacote de conteúdo ou um pacote.

Você pode exibir os pacotes de conteúdo importados na biblioteca.

- Ao expandir um pacote de conteúdo no painel direito, seus objetos filho (que fazem parte da release atual) são exibidos sob o pacote de conteúdo.
- É possível exibir informações sobre as releases mais recentes de uma pasta ou de um pacote de conteúdo em suas propriedades. Clique com o botão direito do mouse no objeto, selecione Propriedades e clique na guia Versões. A guia Trilha de auditoria registra as alterações feitas na pasta ou pacote de conteúdo.

Também é possível exibir todos os pacotes de conteúdo importados na paleta Pacotes de conteúdo da guia Operações.

- Se você clicar no pacote de conteúdo no painel esquerdo, suas propriedades serão exibidas no painel direito.
- Se você clicar nos objetos individuais contidos no painel esquerdo, as propriedades do objeto serão exibidas no painel direito.
- Se o pacote de conteúdo contiver um processo, o lado direito mostrará uma exibição gráfica das instâncias.

### Mais informações

[Versões da release](#) (na página 454)

*Criar um pacote de conteúdo*  
*Adicionar objetos a uma pasta*

Antes de exportar uma pasta como pacote de conteúdo, preencha a pasta com objetos.

Um pacote de conteúdo da release 04.02.00 do CA Process Automation requer que todos os objetos para exportação estejam em uma única pasta. Na release 04.1.00, era possível adicionar objetos como atalhos de pastas diferentes em um pacote para exportação.

Depois de adicionar um objeto, é possível usar o parâmetro Versão ou Versão da release para definir a versão do objeto. Se você tiver personalizado o parâmetro Versão da release, defina a versão do objeto na coluna Versão da release.

É possível definir o atributo de versão da release antes de adicionar uma versão selecionada de cada objeto em uma pasta. Depois de adicionar objetos à pasta, defina a versão da release do pacote de conteúdo.

**Siga estas etapas:**

1. Crie uma pasta.
2. Para abrir a pasta, clique duas vezes nela no painel de pastas do navegador da biblioteca.
3. Na barra de ferramentas, clique em Novo e, em seguida, selecione um tipo de objeto.  
O CA Process Automation cria um objeto.
4. Forneça um nome exclusivo para o objeto.

**Observação:** também é possível recortar e colar um objeto de outra pasta para a nova pasta do pacote de conteúdo.

*Adicionar informações da release a uma pasta*

A adição de informações da release a uma pasta é uma forma útil de fornecer informações relevantes sobre a pasta e quaisquer dos objetos contidos nela.

É possível definir a release da pasta na guia Release nas propriedades da pasta. O valor do atributo ReleaseVersion que você definir para uma pasta na guia Release permite gerenciar a versão de uma pasta, o que é especialmente útil quando você deseja exportar essa pasta como pacote de conteúdo. ReleaseVersion é o atributo obrigatório sobre a pasta. Não é possível excluir ou editar o atributo ReleaseVersion, mas é possível inserir a versão da release apropriada na coluna Valor.

**Observação:** é possível exibir as informações sobre as releases mais recentes de uma pasta nas propriedades da pasta. Clique com o botão direito do mouse na pasta, selecione Propriedades e, em seguida, clique na guia Release. A guia Trilha de auditoria registra todas as alterações feitas na pasta.

**Siga estas etapas:**

1. Clique na guia Biblioteca.
2. Clique com o botão direito do mouse no painel direito e selecione Propriedades.
3. No painel Propriedades, clique na guia Release para exibir as propriedades da pasta.
4. Clique no campo Valor.  
A caixa Valor é exibida.
5. Digite um valor para o atributo ReleaseVersion e, em seguida, clique em OK.
6. Clique no link Adicionar propriedade para definir uma propriedade de pasta adicional e clique no botão Excluir propriedade para remover uma propriedade de pasta definida pelo usuário.

Depois que a versão da release tiver sido atribuída, você poderá exportar a pasta como pacote de conteúdo.

## Mais informações

[Exportar uma pasta](#) (na página 462)

### *Chamar objetos de pacote de conteúdo*

Quando qualquer objeto chama outro objeto no CA Process Automation, o produto recupera o objeto de referência do banco de dados, a menos que o objeto tenha sido excluído. Por exemplo, um processo pode chamar outro processo, ou um processo pode se referir a um determinado conjunto de dados. Se o objeto de referência tiver sido excluído, o produto exibirá um erro. Caso contrário, o produto retornará o objeto.

No entanto, há outra restrição para chamar um objeto que está em um pacote de conteúdo. Se qualquer objeto fizer referência a outro objeto em um pacote de conteúdo, o produto verificará se ele reside na versão da release *atual* do pacote de conteúdo. Se o objeto não estiver na versão da release atual do pacote de conteúdo (ou tiver sido excluído), o produto retornará um erro.

O produto requer a versão mais atual de um objeto em um pacote de conteúdo nas seguintes instâncias:

- Você pesquisa um operador personalizado no navegador do operador.
- Um calendário chama outro calendário para visualizar calendário de exclusões (no campo Visualizar calendário de exclusões, localizado na guia Visualizar do criador de calendários). Se o calendário selecionado não fizer parte da versão da release atual do pacote de conteúdo, suas datas não serão excluídas quando esse calendário for aplicado com o campo Visualizar calendário de exclusões.
- Um operador personalizado chama outro objeto por meio de uma referência ao objeto que não faz parte da versão da release atual do pacote de conteúdo.
- Um conjunto de dados que o operador Iniciar processo usa chama outro objeto por meio de uma referência ao objeto. A referência ao objeto não está na versão da release atual do pacote de conteúdo.
- Um processo usa o operador Verificar o calendário para chamar um calendário que não faz parte da versão da release atual do pacote de conteúdo.
- Uma exibição de processos chama outro objeto que não faz parte da versão da release atual do pacote de conteúdo. Neste exemplo, a paleta Exibição de processos não exibe essa exibição de processos.
- Um processo chama um formulário de solicitação de interação que não está na versão da release atual do pacote de conteúdo. Nessa instância, esse processo falha.
- Um processo chama outro processo que não está na versão da release atual do pacote de conteúdo. Nessa instância, esse processo falha.
- Uma programação chama outro processo que não está na versão da release atual do pacote de conteúdo. Nessa instância, esse processo não é executado.

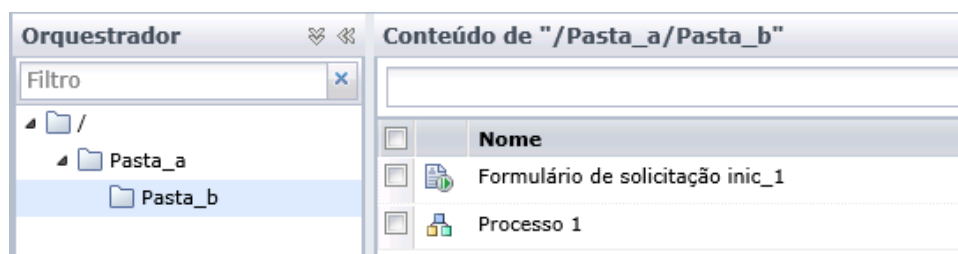
- Um formulário de solicitação inicial chama outro objeto por meio de uma referência ao objeto que não faz parte da versão da release atual do pacote de conteúdo.
- Um formulário de solicitação de interação (usado no operador Atribuir tarefa de usuário e chamado em um operador Iniciar processo) chama outro objeto por meio de uma referência ao objeto que não faz parte da versão da release atual do pacote de conteúdo.

## Exportar caminhos relativos ou Exportar caminhos absolutos

Ao iniciar uma exportação, você pode optar por exportar caminhos absolutos ou exportar caminhos relativos. A opção selecionada afeta a estrutura de pastas na qual o objeto selecionado aparece na biblioteca do orquestrador de destino.

### Exemplo: exportação de um pacote de conteúdo

Considere o exemplo a seguir. Na pasta raiz, a biblioteca do orquestrador de origem é Pasta\_a contém Pasta\_b, que contém o Processo 1 e o Formulário de solicitação inicial 1.



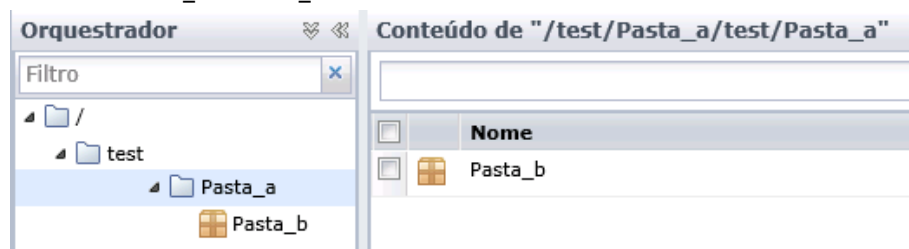
**Caso 1:**

Suponha que os critérios sejam os seguintes:

- Você exportou a Pasta\_b (incluindo o Processo 1 e o Formulário de solicitação inicial 1) como pacote de conteúdo com um caminho absoluto.
- O usuário importador seleciona a pasta /teste como o destino no novo ambiente.

Após a importação, o pacote de conteúdo aparece na seguinte estrutura de pastas:

/teste/Pasta\_a/Pasta\_b



Na hierarquia, teste e Pasta\_a são pastas, mas Pasta\_b agora é um pacote de conteúdo.

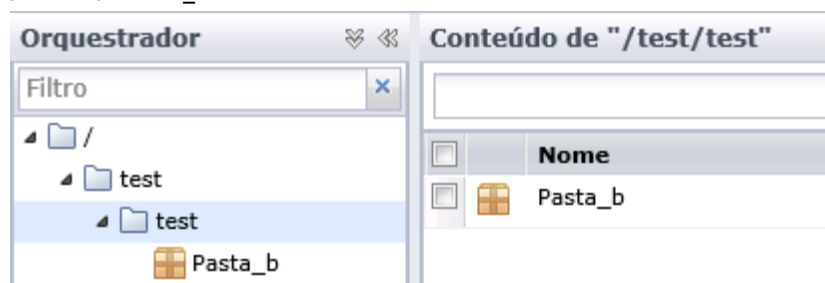
**Caso 2:**

Suponha que os critérios sejam os seguintes:

- Você exportou a Pasta\_b com um caminho relativo.
- O usuário importador seleciona a pasta /teste como o destino.

Após a importação, o pacote de conteúdo aparece na seguinte estrutura de pastas:

/teste/Pasta\_b

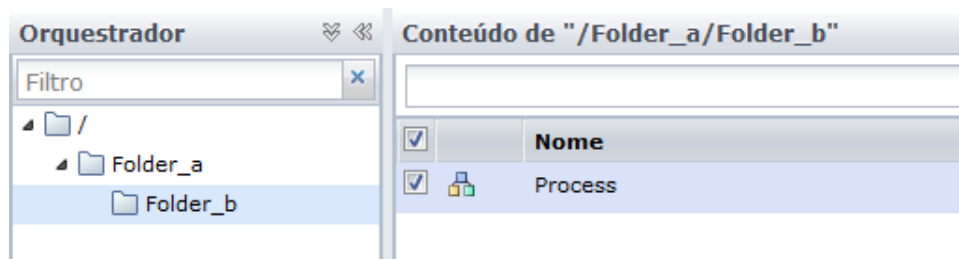


### Exemplo: Exportação de pasta

Suponha que os critérios sejam os seguintes:

- A estrutura de pastas de origem é raiz, Pasta\_a, Pasta\_b.
- O objeto Processo reside na Pasta\_b.

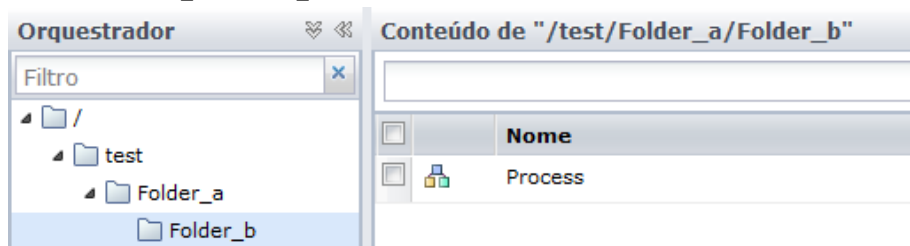
/Pasta\_a/Pasta\_b/Processo



#### Caso 1:

Exportar Pasta\_b com caminhos absolutos. Importar o conteúdo exportado para a pasta /teste. Após a importação, a estrutura no destino será:

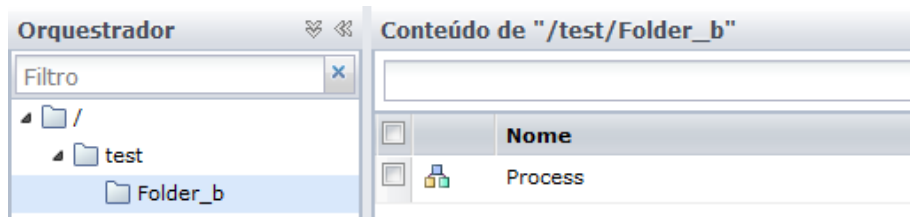
/teste/Pasta\_a/Pasta\_b/Processo



#### Caso 2:

Exportar Pasta\_b com caminhos relativos. Importar o conteúdo exportado para a pasta /teste. Após a importação, a estrutura no destino será:

/teste/Pasta\_b/Processo

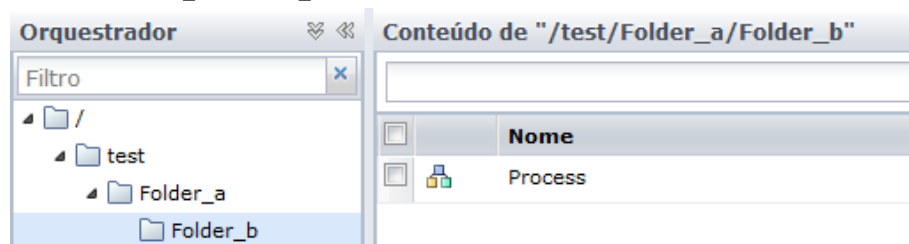


### Exemplo: Exportação de objeto

#### Caso 1:

Exportar o objeto Processo com caminhos absolutos. Importar o conteúdo exportado para a pasta /teste. Após a importação, a estrutura no destino será:

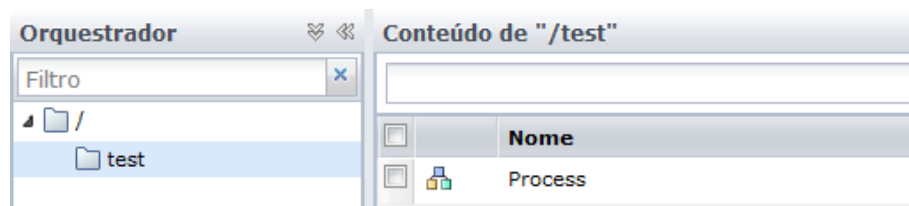
/teste/Pasta\_a/Pasta\_b/Processo



#### Caso 2:

Exportar o objeto Processo com caminhos relativos. Importar o conteúdo exportado para a pasta /teste. Após a importação, a estrutura no destino será:

/teste/Processo



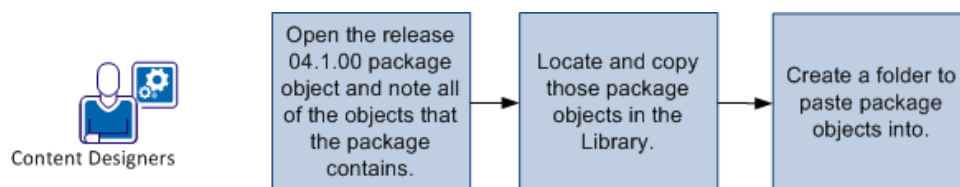
## Como migrar os objetos de um pacote para uma nova pasta para se preparar para a exportação

O pacote de conteúdo é um objeto apresentado no CA Process Automation 04.2.00. No CA Process Automation 04.1.00, os objetos estavam contidos em um pacote para exportação a partir de um ambiente para outro ambiente. O objeto do pacote é suportado somente para compatibilidade com versões anteriores no 04.2.00. Você não pode fazer referência a pacotes em formulários de solicitação de interação, formulários de solicitação inicial, operadores personalizados, exibições de processos ou conjuntos de dados. Você ainda pode exportar um pacote da 04.1.00, mas não pode modificá-lo depois de importá-lo para um novo ambiente. No entanto, é possível modificar seus objetos.

Se desejar agrupar objetos semelhantes e exportá-los como uma *release não modificável* para um novo ambiente, será necessário usar o pacote de conteúdo. O CA Process Automation cria o pacote de conteúdo quando uma pasta é exportada (como pacote de conteúdo) de um ambiente e importada para outro ambiente. Depois de importado, o pacote de conteúdo poderá ser exibido no navegador da biblioteca e na guia Operações, na paleta Pacotes de conteúdo, como indicado pelo ícone do pacote de conteúdo 📦.

Este cenário descreve as etapas necessárias para migrar o conteúdo de um objeto do pacote para uma nova pasta, se desejar continuar movendo o conteúdo como uma release não modificável entre ambientes.


O diagrama a seguir ilustra as tarefas que os criadores de conteúdo executam ao migrar os objetos de um pacote para uma nova pasta para se preparar para a exportação:



### Siga estas etapas:

1. [Abrir o objeto do pacote da release 04.1.00 e anotar todos os objetos contidos no pacote](#) (na página 473).
2. [Localizar e copiar esses objetos do pacote na biblioteca](#) (na página 473).
3. [Criar uma pasta para colar os objetos do pacote](#) (na página 474).

## Abrir o objeto do pacote da release 04.1.00 e anotar todos os objetos contidos no pacote

O objeto do pacote criado na release 04.1.00 é indicado por um ícone de pacote azul: 

### Siga estas etapas:

1. Abra o Navegador da biblioteca.
2. Localize e abra o objeto do pacote da release 04.1.00.
3. Anote todos os objetos contidos no pacote.

## Localizar esses objetos do pacote na biblioteca

Depois de identificar os objetos no pacote, localize-os na biblioteca, copie-os e cole-os em uma nova pasta.

Você pode abrir e percorrer pastas conhecidas da biblioteca ou pesquisar nas pastas pelo nome dos objetos que o pacote contém. Além disso, você pode pesquisar objetos com a Pesquisa básica, usando um único critério, ou com a Pesquisa avançada, usando vários critérios. A pesquisa localiza os objetos que correspondem aos seus critérios de pesquisa na estrutura de pastas que você especificar.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Navegue pela árvore de pastas até a pasta que contém o objeto. Conforme necessário, é possível consultar as pastas por nome usando o campo de pesquisa sobre o painel do orquestrador.
3. (Opcional) Execute uma pesquisa pelo objeto. Na pasta selecionada, execute um dos seguintes tipos de pesquisa:
  - Use uma pesquisa básica se consultar com um único critério de pesquisa, como tipo de objeto ou proprietário.
  - Use uma pesquisa avançada se consultar com vários critérios, como estado do objeto ou data de modificação.
4. Clique com o botão direito do mouse em cada objeto que a pesquisa retornar e selecione Copiar.

## Criar uma pasta para colar os objetos do pacote

Uma vez que tiver localizado um objeto na biblioteca que o pacote da release 04.1.00 contém, cole-o em uma nova pasta para exportá-lo para um novo ambiente.

Todos os objetos que deseja exportar devem estar localizados em apenas uma pasta. Na release 04.1.00, era possível adicionar objetos como atalhos de pastas diferentes em um pacote para exportação. Esse comportamento não é possível com um pacote de conteúdo. Todos os objetos devem estar armazenados em apenas uma pasta para exportação.

### Siga estas etapas:

1. No painel de pastas do navegador da biblioteca, selecione a pasta na qual deseja criar uma pasta.
2. Na barra de ferramentas, clique em Novo e, em seguida, clique em Pasta.  
A nova pasta é exibida.
3. Renomear a pasta.
4. Clique com o botão direito do mouse na pasta e, em seguida, selecione Colar.

Repita o processo para cada objeto que o pacote contiver até que a nova pasta contenha todo o conteúdo da release 04.1.00. Você pode editar os objetos quando forem colados na nova pasta.

A migração de conteúdo foi concluída e você pode exportar a pasta para um novo ambiente.

## Importando

Os tópicos nesta seção explicam o processo de importação.

## Como definir opções de importação

O CA Process Automation fornece certa flexibilidade para importar objetos.

**Se um objeto importado tiver o mesmo nome que um objeto existente:**

Importar

Importar

Não importar

Importar e substituir

Definir a versão importada como atual

Tornar os operadores personalizados disponíveis

Publicar a configuração do grupo do operador personalizado

Se a importação incluir operadores personalizados, selecione Tornar os operadores personalizados disponíveis.

Se os operadores personalizados forem novos e pertencerem a um novo grupo personalizado, execute a ação apropriada para o seu ambiente.

- Não selecione Publicar a configuração do grupo do operador personalizado se o ambiente de importação estiver no mesmo domínio do ambiente de exportação. Nesse caso, a configuração do grupo de operadores personalizados já foi publicada.
- Selecione Publicar a configuração do grupo do operador personalizado se o ambiente de importação estiver em um domínio diferente do ambiente de exportação

Considere o conteúdo de importação quando você configurar Definir a versão importada como atual e selecionar como lidar com nomes duplicados.

- Para ativar os objetos importados, com a possibilidade de reverter para uma versão anterior de um objeto importado, se necessário:
  - Selecione: Importar
  - Selecione: Definir a versão importada como atual.

**Observação:** essas opções são melhores quando você estiver importando uma versão da release de atualização e todos os destinos do operador estiverem configurados como hosts no ambiente de importação. Você pode esperar ser notificado sobre nomes duplicados porque os objetos da última release estão localizados na pasta de destino.

- Para importar sem ativar os objetos atualizados, onde a versão anterior mantém seu status da versão atual:
  - Selecione: Importar
  - Desmarque: Definir a versão importada como atual

**Observação:** essas opções são melhores quando a importação incluir operadores que usam hosts de destino que ainda não estão definidos com seu nome de touchpoint no ambiente de importação. Com essa configuração, é possível tornar os objetos atuais depois de verificar se os destinos de processo estão disponíveis no ambiente de importação.

- Para adiar a importação de qualquer objeto com um nome duplicado e optar por tornar os objetos atuais manualmente:
  - Selecione: Não importar
  - Desmarque: Definir a versão importada como atual

– **Observação:** essas opções são melhores quando você estiver importando novos objetos para uma pasta preenchida. Essas opções evitam que um objeto de importação se torne uma nova versão de um objeto com o mesmo nome, mas com uma função diferente. Essas opções também permitem tornar os objetos atuais depois de testar e verificar o respectivo uso no novo ambiente.

Se você receber alertas, considere estas ações:

- Registre os nomes duplicados na mensagem de alerta e informe um administrador no ambiente de origem. Talvez esses objetos possam ser renomeados e exportados novamente.
  - Importe novamente, mas importe para uma pasta vazia.
- Para ativar os objetos importados sem a possibilidade de reverter a ação para os objetos com nomes duplicados:
    - Selecione: Importar e substituir
    - Selecione: Definir a versão importada como atual.
    - **Observação:** essas opções são melhores quando você estiver importando novamente correções de objetos para a pasta de destino. Nesse caso, você nunca precisará reverter para a versão substituída.

## Executar uma importação

É possível importar um objeto, uma pasta, um pacote ou um pacote de conteúdo que tenha sido exportado. O processo de exportação cria o arquivo XML que o processo de importação utiliza.

**Observação:** se você exportar uma pasta como pacote de conteúdo, o produto importará a pasta exportada como pacote de conteúdo não modificável na pasta de importação. Não é possível recortar e excluir de um pacote de conteúdo. Você pode copiar os objetos de automação de um pacote de conteúdo e colá-los em outra pasta para modificá-los. Você pode visualizar todos os pacotes de conteúdo importados na paleta Pacotes de conteúdo da guia Operações.

### Siga estas etapas:

1. Clique na guia Biblioteca.
2. Clique em Orquestrador e, em seguida, selecione o *Orquestrador:ambiente* apropriado.
3. Vá até a pasta de destino.
4. Clique com o botão direito do mouse na pasta de destino e, em seguida, selecione Importar.
5. Na caixa de diálogo Importar, vá até o arquivo XML exportado e, em seguida, clique em Abrir.

**Observação:** certifique-se de que o arquivo XML que deseja importar não é maior que 250 MB.

6. Especifique como importar um objeto que possui o mesmo nome de um objeto existente no mesmo caminho.

**Observação:** se você importar para uma pasta vazia, não haverá conflitos com nomes existentes.

**Importar**

Processa o objeto importado como uma nova versão do objeto existente. Selecione essa opção se a finalidade da importação for uma atualização e você desejar manter o histórico de versões anteriores. Se o objeto importado tiver a mesma versão da release, o produto substituirá a existente pela versão da release do objeto importado.

**Não importar**

Interrompe a importação do objeto e mantém o objeto existente. O processo de importação lista os objetos com nomes conflitantes. Se houver conflitos, importe para uma pasta vazia. Como opção, é possível renomear o objeto no ambiente de origem e, em seguida, repetir a exportação e a importação. Essa opção é ótima quando os objetos que estão sendo importados forem objetos novos em vez de novas versões de objetos existentes.

**Importar e substituir**

Exclua o objeto existente e importe uma nova versão do objeto como versão 0.

7. Execute as especificações de importação de acordo com as seguintes diretrizes:

#### **Definir versão importada como atual**

Especifica se é preciso definir a versão importada como atual. Esta opção se aplica se você selecionar Importar e Substituir na etapa 6.

- **Selecionado:** torna a versão importada a versão atual.
- **Desmarcado:** mantém como atual a versão existente que está marcada como Atual. Desmarque essa opção se a versão anterior desse objeto de dados estiver ativa.

**Observação:** se você pesquisar em uma hierarquia de pastas com um pacote de conteúdo, o produto retornará apenas os objetos na versão da release atual do pacote de conteúdo. O produto também retorna objetos armazenados na hierarquia de pastas, mas que não estão incluídos no pacote de conteúdo.

#### **Tornar os operadores personalizados disponíveis**

Especifica se é preciso definir os operadores personalizados importados como Disponível. Não é possível usar operadores personalizados até que eles estejam disponíveis.

- **Selecionado:** define os operadores personalizados importados como disponíveis. Selecione essa opção para que os usuários não precisem tornar disponíveis individualmente cada operador personalizado importado.
- **Desmarcado:** importar operadores personalizados sem alterar sua disponibilidade.

#### **Publicar a configuração do grupo do operador do cliente**

Especifica se você deseja publicar grupos de operadores personalizados. Publica ao importar o objeto, a pasta ou o pacote para um domínio diferente daquele a partir do qual o arquivo foi exportado. A publicação publica os grupos de operadores personalizados da guia Módulos no nível do domínio, em que cada ambiente contém também os grupos herdados do domínio.

- **Selecionado:** publica os grupos de operadores personalizados. Essa configuração se aplica quando você importar grupos de operadores personalizados para um domínio diferente do domínio no qual os grupos de operadores personalizados foram criados.
- **Desmarcado:** não publica os grupos de operadores personalizados. Essa configuração se aplica quando as configurações do grupo importado de operadores personalizados foram publicadas para o domínio atual e seus ambientes, quando o grupo de operadores personalizados foi definido.

**Observação:** se você não tiver direitos Group\_Config\_Admin, essa opção estará desativada.

8. Clique em Enviar.
9. Clique em OK na mensagem de confirmação de importação bem-sucedida.

## Versão da release e Status da definição da linha de base de objetos importados

O estado do atributo Versão da release (modificável ou não modificável) e o Status da definição da linha de base de cada objeto importado dependem das seguintes condições:

- Se o ambiente de importação já contiver versões existentes dos objetos que estão sendo importados.
- Se os objetos existirem, se eles forem importados com a opção para importar como uma nova versão e mantiverem as versões existentes do objeto.

As seguintes opções de importação estão disponíveis para a manipulação de objetos existentes nas quais a importação é uma nova versão:

- Excluir o objeto existente e tratar a versão do objeto importado versão como a versão 0.

**Se um objeto importado tiver o mesmo nome que um objeto existente:**

Importar e substituir

Definir a versão importada como atual

- Manter o objeto existente e tratar a versão do objeto importado como uma nova versão.

**Se um objeto importado tiver o mesmo nome que um objeto existente:**

Importar

### Caso 1: as versões importadas são as únicas versões dos objetos no ambiente de importação.

A versão importada é a única versão do objeto nas seguintes situações:

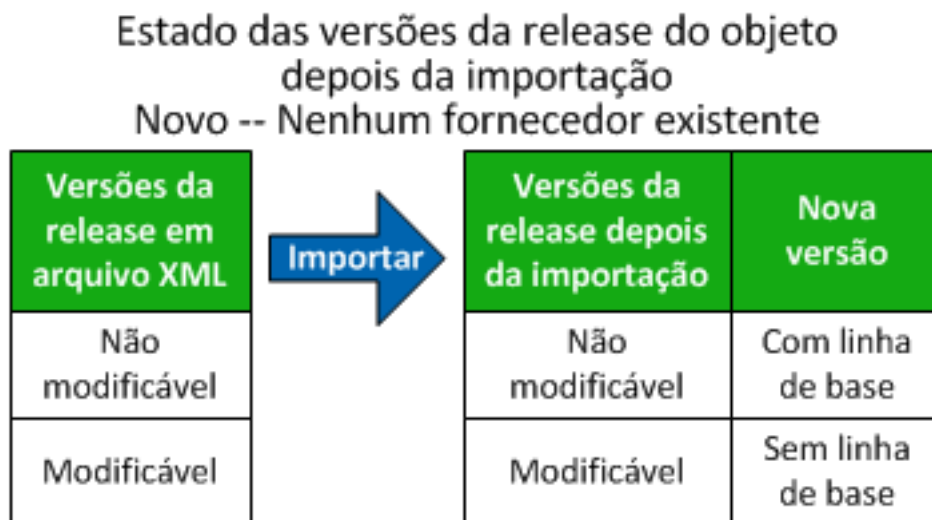
- O objeto não existia no ambiente de importação antes de ser importado.
- O objeto existia, mas o objeto importado substituiu o objeto existente com o mesmo nome.

Considere as propriedades de um objeto que você criar. A versão 0 do objeto é marcada como atual, não é definida como linha de base e possui um atributo Versão da release.



- Quando a versão da release do arquivo XML for não modificável, o estado do atributo Versão da release do objeto importado será não modificável e a nova versão será definida como linha de base.
- Quando a versão da release do arquivo XML for modificável, o estado do atributo Versão da release do objeto importado será modificável e a nova versão não será definida como linha de base.

A ilustração a seguir resume os casos:



**Caso 2: as versões importadas são as novas versões de objetos existentes no ambiente de importação.**

Considere o caso em que você importa uma nova versão de um objeto existente, em que o atributo Versão da release do objeto existente é não modificável.

- Quando a versão da release do arquivo XML for não modificável, o estado do atributo Versão da release do objeto importado será não modificável e a nova versão será definida como linha de base.
- Quando a versão da release do arquivo XML for modificável, o estado do atributo Versão da release do objeto importado será não modificável e a nova versão não será definida como linha de base.

Considere o caso em que você importa uma nova versão de um objeto existente, em que o atributo Versão da release do objeto existente é modificável.

- Quando a versão da release do arquivo XML for não modificável, o estado do atributo Versão da release do objeto importado será não modificável e a nova versão será definida como linha de base.
- Quando a versão da release do arquivo XML for modificável, o estado do atributo Versão da release do objeto importado será modificável e a nova versão não será definida como linha de base.

A ilustração a seguir resume estes casos:

**Estado das versões da release do objeto antes e depois da importação  
Importar como uma nova versão e manter as versões existentes**

Versões da release antes da importação	Versões da release em arquivo XML	Importar	Versões da release depois da importação	Nova versão
Não modificável	Não modificável		Não modificável	Com linha de base
Não modificável	Modificável		Não modificável	Sem linha de base
Modificável	Não modificável		Não modificável	Com linha de base
Modificável	Modificável		Modificável	Sem linha de base

## Valores mantidos na versão inicial de um objeto copiado

A *versão atual* de um objeto é o assunto de uma ação copiar e colar. Ou seja, a nova cópia tem uma versão, a versão atual do objeto original. O valor Versão da release e se ela é modificável são mantidos na versão inicial da cópia do objeto. Se na importação a versão atual do objeto original foi definida como linha de base, a cópia do objeto também será definida como linha de base.



# Apêndice A: Formatar especificadores para datas

---

O CA Process Automation usa a sequência de caracteres padrão de formato de data e hora do Java padrão. Na sequência de caracteres padrão de data e hora, letras entre aspas de 'A' a 'Z' e de 'a' a 'z' são interpretadas como letras padrões que representam os componentes de uma sequência de caracteres de data ou hora. O texto pode ser colocado entre aspas usando aspas simples (') ou aspas duplas (") para evitar a interpretação. "" representa uma aspa simples. Todos os outros caracteres não são interpretados; eles são copiados para a sequência de caracteres de saída durante a formatação ou comparadas contra a sequência de caracteres de entrada durante a análise.

Letra	Componente de data ou hora	Apresentação	Exemplos
l	Designador era	Texto	AD
a	Ano	Ano	2009; 09
m	Mês do ano	Mês	Julho; 07
w	Semana do ano	Número	27
W	Semana do mês	Número	2
D	Dia do ano	Número	189
d	Dia do mês	Número	10
F	Dia da semana do mês	Número	2
E	Dia da semana	Texto	Terça-feira; ter
a	Marcador AM/PM	Texto	PM
h	Hora do dia (0-23)	Número	0
k	Hora do dia (1-24)	Número	24
K	Hora em AM/PM (0-11)	Número	0
h	Hora em AM/PM (1-12)	Número	12
M	Minuto da hora	Número	30
s	Segundo do minuto	Número	55
s	Milissegundo	Número	978
z	Fuso horário	Fuso horário geral	Hora Padrão do Pacífico; PST; GMT-08:00

Letra	Componente de data ou hora	Apresentação	Exemplos
Z	Fuso horário	Fuso horário RFC 822	-0800

Padrões geralmente são repetidos; seu número determina a apresentação exata como mostrado na tabela a seguir:

Apresentação	Formatação
Texto	Se a quantidade de caracteres padrões é 4 ou mais, todo o formulário é usado; caso contrário, uma forma curta ou abreviada é aceita, independente do número de caracteres padrões.
Número	A quantidade de caracteres padrão é o número mínimo de dígitos e números menores são analisados em zero para esse valor. Durante a análise o número de caracteres padrões é ignorado, a menos que seja necessário para separar dois campos adjacentes.
Ano	Se a quantidade de letras padrões é 2, o ano é truncado para 2 dígitos; caso contrário, ele é interpretado como um número.  Durante a análise, se o número de caracteres padrões é maior do que 2, o ano é interpretado literalmente, independentemente do número de dígitos. Usando o padrão "mm/dd/aaaa", "01/11/12" analisa para 11 de janeiro de 2012 D.C.
Mês	Se a quantidade de caracteres padrões é 3 ou mais, o mês é interpretado como texto; caso contrário, é interpretado como um número.
Fuso horário geral	Os fusos horários são interpretados como texto se eles têm nomes. Para fusos horários que representam um valor de desvio do GMT, a sintaxe a seguir será usada: <ul style="list-style-type: none"> <li>■ GMTOffsetTimeZone:</li> <li>■ Horas do símbolo de GMT: minutos</li> <li>■ O sinal é + ou -</li> <li>■ As horas devem ser um valor entre 0 e 23 e um dos seguintes formatos: Dígito Dígito Dígito</li> <li>■ Os minutos devem estar entre 00 e 59 e no seguinte formato: Dígito Dígito</li> <li>■ Dígito é um dos seguintes: 0 1 2 3 4 5 6 7 8 9</li> </ul> O formato é independente do local e deve ser assumido do bloco de latim básico do Unicode padrão.

Apresentação	Formatação
Fuso horário RFC 822	Um formato de fuso horário de quatro dígitos é usado: <ul style="list-style-type: none"> <li>■ RFC822TimeZone: Sing TwoDigitHours Minutes</li> <li>■ TwoDigitHours deve estar entre 00 e 23. As outras definições são como fusos horários gerais.</li> </ul>

SimpleDateFormat também oferece suporte a sequências de caracteres padrões de data e hora localizada. Nessas sequências de caracteres, os caracteres padrões descritos acima podem ser substituídos por outro, dependendo do local e caracteres padrão. SimpleDateFormat não lida com a localização do texto além dos caracteres padrões; que é até o cliente da classe.

Os exemplos a seguir mostram como os padrões de data e hora são interpretados nos inglês dos estados unidos local. A data e a hora específicas são 2009-07-04 12:08:56 no fuso horário do Pacífico dos Estados Unidos.

Padrão de data e hora	Resultado
"aaaa.MM.dd G 'at' HH:mm:ss z"	2009.07.04 AD às 12:08:56 PDT
"EEE, MMM d, aa"	Qua, 4 de jul de 09
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 horas, horário de verão do Pacífico
"K:mm a, z"	0:08 PM, PDT
"aaaa.MMMMMM.dd GGG hh:mm aaa"	02009.Julho.04 AD 12:08 PM
"EEE, d MMM aaaa HH:mm:ss Z"	Qua, 4 Jul 2009 12:08:56 -0700
"aaMMddHHmmssZ"	010704120856-0700
"aaaa-MM-dd'T'HH:mm:ss.SSSZ"	2009-07-04T12:08:56.235-0700



# Apêndice B: Usando máscaras para especificar padrões nas sequências de caracteres

---

As máscaras são usadas para parâmetros do operador que monitoram as mensagens e o conteúdo do arquivo. As máscaras usadas pelo CA Process Automation são expressões regulares do JavaScript que descrevem conjuntos de sequências de caracteres sem realmente listar o seu conteúdo.

As máscaras podem ser usadas para determinar se um padrão específico particular é correspondido e para extrair as partes da sequência de caracteres correspondente. O CA Process Automation coloca subsequências de caracteres extraídas nas variáveis que podem ser usadas para especificar valores de parâmetros para outros operadores. Operadores que usam máscaras são os operadores de Entradas do diretório e macros de extração. Também é possível usar as máscaras para especificar sequências de caracteres permitidas, como formatos de data para campos de entrada de texto.

Esta seção contém os seguintes tópicos:

[Sintaxe da máscara](#) (na página 489)

[Máscara de exemplo](#) (na página 493)

## Sintaxe da máscara

Uma expressão regular consiste em uma sequência de caracteres que deve corresponder aos tipos que aparecem nas sequências de caracteres correspondentes. Os caracteres exibidos na tabela a seguir servem como operadores de expressão regular

Operador de expressão regular	Descrição	Exemplo:
.	O operador ponto representa qualquer caractere, exceto um caractere de final de linha.	A expressão a.b corresponde a qualquer sequência de três caracteres começando com "a" e terminando com "b".

Operador de expressão regular	Descrição	Exemplo:
\	<p>O escape permite o uso do operador como caracteres literais em vez de ser reconhecido como um operador.</p> <p>Se uma máscara é especificada como uma sequência de caracteres literal em uma expressão do CA Process Automation, o próprio operador de escape deve ser precedido por caracteres de escape para evitar que o caractere de escape seja eliminado pelo intérprete de expressão do CA Process Automation antes da máscara ser passada para o analisador de expressão regular.</p>	<p>O caractere de escape \\ coincide com o caractere de barra invertida \.</p> <p>O operador de escape \  corresponde a barra vertical.</p> <p>Para uma sequência de caracteres literal em uma expressão do CA Process Automation, "\\?" corresponde ao ponto de interrogação.</p>
\n	<p>Sequência de caracteres de escape do fim de linha.</p> <p>Se uma máscara é especificada como uma sequência de caracteres literal em uma expressão do CA Process Automation, a sequência de caractere de escape de fim de linha deve ser precedida por caracteres de escape para evitar que o caractere de escape seja eliminado pelo intérprete de expressão do CA Process Automation antes da máscara ser passada para o analisador de expressão regular.</p>	<p>A expressão primeira linha\nsegunda linha corresponde as sequências de caracteres onde a “primeira linha” será exibida no final de uma linha e a “segunda linha” será exibida no início da próxima linha.</p> <p>Se a mesma expressão é especificada por uma sequência de caracteres literal em uma expressão do CA Process Automation, o operador é escapado como segue: "primeira linha\\nsegunda linha"</p>
	<p>Alternativa. Esse operador cria uma alternativa entre os caracteres ou grupos. Ele se aplica ao caractere ou grupo imediatamente adjacente a ele a sua esquerda ou direita.</p>	<p>(a b) corresponde o caractere “a” ou o caractere “b”.</p> <p>(abcd) (efgh) corresponde à sequência de caracteres “abcd” ou “efgh”.</p> <p>(ab Bc) corresponde a sequência de caracteres “abc” ou a sequência de caracteres “aBc”.</p>
()	<p>Grupo. Esse operador de caracteres em grupo em uma sub-expressão que se aplica a operadores de alternativa e de repetição.</p>	<p>(abcd) (efgh) corresponde à sequência de caracteres “abcd” ou “efgh”.</p> <p>(abcde)* corresponde a zero, um ou várias ocorrências sucessivas da sequência de caracteres “abcde”, onde abcde* corresponde a sequências de caracteres começando com “abcde” seguido por um zero ou mais caracteres “e”.</p>

Operador de expressão regular	Descrição	Exemplo:
[ ]	Usado para definir uma classe de caracteres por inclusão. Ele é composto por uma lista de caracteres individuais e intervalos (classes).	Essa construção corresponde a um único caractere. [0-9] corresponde a um dígito. [aeiou] corresponde a uma vogal. [a-cx-z9] corresponde a uma letra entre os três primeiros e os três últimos caracteres do alfabeto ou o dígito 9. [0-9] + corresponde a um número inteiro. [^] Usado para definir uma classe de caracteres por exclusão. Essa construção corresponde a um único caractere. [^0-9] corresponde a qualquer caractere, exceto um dígito. [^aeiou] corresponde a qualquer caractere, exceto uma vogal.
*	Repetição. Define zero ou mais ocorrências do caractere ou grupo exibido ao lado do operador. A série pode ser deixada em branco. a* corresponde a zero ou mais caracteres "a".	(123)* corresponde a zero ou mais repetições da sequência de caracteres "123". 123* corresponde ao dígito 12 seguido por zero ou mais caracteres "3".
+	Repetição. Define uma ou mais ocorrências do caractere ou grupo exibido ao lado do operador. A série não pode ser vazia.	a+ corresponde a um ou mais caracteres "a". (ab)+ corresponde a uma série de uma ou mais repetições da sequência de caracteres "ab". [a-z,  .] corresponde a uma série de palavras em caracteres pequenos, separados por espaços em branco, vírgulas ou períodos.
?	Usado para indicar que o caractere ou grupo que precede o operador é opcional. ab?c corresponde à sequência de caracteres "abc" e "ac".	a(X Y)?b corresponde à sequência de caracteres "ab", "aXb" e "aYb".

<b>Operador de expressão regular</b>	<b>Descrição</b>	<b>Exemplo:</b>
{n}	Usado para definir um número preciso de repetições do caractere ou grupo que precede o operador.	[0-9]{2} corresponde a um número feito de exatamente dois dígitos. (ab){2} corresponde à sequência de caracteres "abab". abc{2} corresponde à sequência de caracteres "abcc". (a{2})* corresponde à sequência de caracteres feita de um número par de "a" caracteres.
{n,m}	Usado para especificar o número de ocorrências do caractere ou grupo que precede o operador é entre n e m.	[0-9]{2,3} corresponde a um número feito de dois ou três dígitos. (ab){1,2} corresponde à sequência de caracteres "ab" ou "abab".
{n,}	Usado para especificar o número de ocorrências do caractere ou grupo que precede o operador é n ou mais.	0-9]{2,} corresponde a um número feito de pelo menos dois dígitos.
{,n}	Usado para especificar o número de ocorrências do caractere ou grupo que precede o operador é n ou menos.	[0-9]{,2} corresponde a um número igual ou menor que 99, incluindo a sequência de caracteres vazia.

## Máscara de exemplo

O seguinte padrão permite que a filtragem precisa da saída do comando DIR no shell de comando do Windows:

```
(([0-9]{2}/){2}[0-9]{2} * [0-9]{2}:[0-9]{2}(a|p) * [0-9,]+ [_~0-9a-zA-Z]*(\.[_~0-9a-zA-Z]*)?)
```

As linhas a seguir correspondem a esse padrão:

```
10/26/05 07:03p      134,217,723 pagefile.sys
```

```
10/26/05 07:03p      1,024 testfile
```

Essas linhas são compostas dos seguintes elementos:

- `(([0-9]{2}/){2}[0-9]{2}` representa as sequências de caracteres compostas de dois dígitos e uma barra (/), como em “10/”.
- `([0-9]{2}/){2}[0-9]{2}` baseia-se no elemento anterior e especifica que a sequência de dois dígitos pode ser repetida três vezes, cada sequência sendo separada da próxima por uma barra, como em “10/26/05”.
- `*[0-9]{2}:[0-9]{2}(a|p)` corresponde a qualquer número de espaços seguidos pela hora no formato hh:mm seguido por “a” para AM ou “p” para PM.
- `*[0-9,]+` corresponde a qualquer número de espaços seguidos por qualquer número de dígitos e vírgulas para o tamanho do arquivo.
- `[_~0-9a-zA-Z]*(\.[_~0-9a-zA-Z]*)?` Representa uma sequência de caracteres alfanuméricos, sublinhados (`_`) e sinais de til (`~`), opcionalmente seguidos por um período e outra sequência. Como o período (`.`) é o operador ponto, é necessário sair usando o caractere de escape (`\`) quando pretendemos correspondê-lo como um caractere.

O padrão em uma máscara pode ser diferente, dependendo dos caracteres exatos a serem correspondidos. Por exemplo `[_~0-9a-zA-Z]*` em vez de `[_~0-9a-zA-Z]*(\.[_~0-9a-zA-Z]*)?` também pode corresponder aos nomes de arquivo em nossas saídas de exemplo, mas sem impor quaisquer restrições no número de períodos (`.`) no nome do arquivo.



# Apêndice C: Como os destinos de um operador são processados

---

Este apêndice descreve como o CA Process Automation processa os destinos do operador.

Esta seção contém os seguintes tópicos:

- [Como os destinos para um operador podem ser especificados](#) (na página 495)
- [Processando um destino especificado como um endereço IP ou FQDN](#) (na página 496)
- [Processando um destino especificado como a ID de um agente ou orquestrador](#) (na página 501)
- [Caso de Uso: rastrear processos recuperados por meio dos logs](#) (na página 502)
- [Exemplo da Recuperação automática dos Operadores](#) (na página 503)

## Como os destinos para um operador podem ser especificados

O destino de um operador é o host no qual o operador é executado. Um host que pode ser um destino de um operador deve ser configurado com um touchpoint, um touchpoint do proxy ou um grupo de hosts. Ao determinar as configurações de execução para um operador, os criadores de conteúdo normalmente se baseiam na seleção do host de destino em como destino é configurado.

Especifique o destino como indicado na lista a seguir quando as condições apropriadas forem atendidas:

### **Touchpoint**

O host de destino tem um orquestrador ou um agente instalado. Quando você especifica um touchpoint como o destino de um operador e esse touchpoint está ativado e associado a um orquestrador ou agente ativo, o CA Process Automation executa o operador no touchpoint especificado. Se um touchpoint de destino estiver associado a apenas um agente ou um orquestrador, cada execução terá como destino o mesmo host. Se um touchpoint de destino estiver associado a vários agentes com a mesma prioridade, cada execução poderá ter como destino um agente diferente em um host diferente.

### **Grupo de touchpoints**

O operador deve ser executado em todos os hosts associados a touchpoints no grupo de touchpoints especificado.

#### <AgentID>

*Exceção para o touchpoint:* especifique a mesma ID do agente para operadores consecutivos quando o processo exigir que o operador seja executado no mesmo host que o operador anterior. Defina uma ID do agente específica quando o processo exigir que o operador seja executado sempre no mesmo host. Uma execução de teste do operador exibe a sequência de caracteres da ID do agente no conjunto de dados do sistema. Copie a sequência de caracteres da ID do agente do conjunto de dados no campo Destino.

#### Touchpoint do proxy

O host de destino é especificado como o host remoto em uma configuração do touchpoint do proxy. Normalmente, esse host não tem um agente ou um orquestrador instalado.

#### Endereço IP ou FQDN.

O host de destino tem um endereço IP ou FQDN que pode ser correspondido com um padrão em um grupo de hosts configurado. Normalmente, esse host não tem um agente ou um orquestrador instalado.

**Observação:** o nome do host pode ser usado, mas FQDN é recomendado.

**Importante:** Podem ocorrer resultados inesperados se você especificar o destino com um touchpoint que tenha o mesmo nome de um host no ambiente. Também podem ocorrer resultados inesperados se o agente associado a esse touchpoint estiver inativo ao executar o operador. Nesse caso, o CA Process Automation não executa o operador no touchpoint. Em vez disso, o CA Process Automation tenta resolver o destino (deve ser o touchpoint) para o nome de um host no ambiente. Se encontrar um nome do host, o CA Process Automation pesquisa outro touchpoint, orquestrador, touchpoint do proxy ou grupo de hosts que executa seus operadores nesse host. O CA Process Automation executa o operador na primeira ocorrência encontrada.

## Processando um destino especificado como um endereço IP ou FQDN

Quando você usa um endereço IP ou FQDN para especificar o host de destino de um operador, o CA Process Automation consulta repetidas vezes o DNS (servidor de nome do domínio). Se especificar um FQDN como o destino, a primeira pesquisa retorna os endereços IP definidos no host com esse FQDN. Se especificar um endereço IP, a primeira pesquisa retorna o FQDN e os endereços IP do host associados a esse FQDN. Cada pesquisa subsequente faz testes para comparar todos os especificadores do host recuperados na pesquisa inicial.

O CA Process Automation procura por todas as formas em que o host de destino possa estar configurado no ambiente do CA Process Automation atual. Um grupo de hosts define os hosts remotos com a sub-rede e os padrões de nome de host. Portanto, um grupo de hosts pode incluir os hosts com os orquestradores ou agentes que são mapeados para touchpoints. Um grupo de hosts pode incluir também hosts individuais que são mapeados para um touchpoint do proxy. Um host identificado com um endereço IP ou FQDN poderia ser processado com um nome de touchpoint ou um nome de touchpoint do proxy.

Algumas pesquisas encontram qualquer orquestrador ou agente ativo definido no host que ativou os touchpoints. Outra pesquisa encontra qualquer touchpoint do proxy ativado em um agente ativo que é mapeado para o host. A pesquisa encontra grupos de hosts ativados em um agente ativo com padrões de nome de host ou sub-redes de endereço IP que correspondem ao FQDN ou endereço IP. Quando os critérios de pesquisa são atendidos, essas consultas atrasam o início da execução do operador. O CA Process Automation executa o operador no primeiro touchpoint ativado encontrado, touchpoint do proxy ou grupo de hosts que está sendo executado em um orquestrador ou agente ativo. O touchpoint, touchpoint do proxy ou grupo de hosts deve pertencer ao ambiente atual. Quando nenhum dos critérios de pesquisa forem atendidos, o operador falhará e uma mensagem de erro será gerada.

A área Destino na guia Configurações de execução para operadores contém uma caixa de seleção Fazer correspondência do destino apenas em grupos de hosts. Marque essa caixa de seleção nos seguintes casos:

- Ao especificar um endereço IP dentro de uma sub-rede especificada em um grupo de hosts.
- Ao especificar um nome de host que corresponda a um padrão especificado em um grupo de hosts.

Ao marcar a caixa de seleção Pesquisar DNS, o CA Process Automation resolve o nome de destino por meio de uma pesquisa no DNS. A pesquisa é para um registro que corresponde a uma referência por um grupo de hosts.

Considere o caso em que você desmarca essa caixa de seleção, mas o endereço IP ou FQDN de destino existe. O seguinte processamento determina o host em que o operador será executado:

1. O aplicativo procura o DNS de todos os identificadores do destino especificado. Se o destino estiver especificado como um FQDN, o CA Process Automation pesquisará todos os endereços IP desse FQDN. Se o destino for um endereço IP, o CA Process Automation pesquisará o FQDN e outros endereços IP para esse FQDN.
  - Se a conexão com o DNS falhar, o aplicativo gravará um erro no arquivo de log e será fechado.
  - Se a consulta retornar o FQDN e outros endereços IP, o aplicativo usará o endereço IP especificado nesse conjunto de resultados em todas as pesquisas subsequentes.
  - Se a consulta retornar um ou mais endereços IP, o aplicativo usará o FQDN especificado e os endereços IP em todas as pesquisas subsequentes.
2. O aplicativo procurará um orquestrador ativo com um touchpoint ativado (ambiente atual) instalado em um host nos resultados da pesquisa da Etapa 1.
  - Se encontrar, o aplicativo executará o operador nesse touchpoint do orquestrador.
  - Se não for encontrado, o aplicativo continuará o processamento.  
**Observação:** o aplicativo não detecta orquestradores agrupados. Os hosts com orquestradores agrupados não aparecem na configuração de domínio.
3. O aplicativo procurará um agente ativo com um touchpoint ativado (ambiente atual) instalado em um host nos resultados da pesquisa da Etapa 1.
  - Se o host tiver um agente ativo com um touchpoint ativado, o CA Process Automation usará esse touchpoint para executar o operador.
  - Se o host tiver um agente ativo com vários touchpoints ativados, o CA Process Automation usará um dos touchpoints ativados para executar o operador.
  - Se não for encontrado, o aplicativo continuará o processamento.

4. O aplicativo procurará um agente ativo que estiver conectado ao host remoto de destino. Esse agente deve ter um touchpoint do proxy ativado ou um grupo de hosts no ambiente atual. Para grupos de hosts, as consultas usam a correspondência de padrão. As consultas do grupo de hosts descobrem se a sub-rede referenciada ou se o padrão de nome de host inclui um identificador de host no conjunto de resultados da Etapa 1. Para touchpoints do proxy, o aplicativo pesquisa uma correspondência exata para qualquer um dos identificadores de host recuperados na Etapa 1.
  - Se a única correspondência for um host remoto referenciado por um touchpoint do proxy, o aplicativo executará o operador nesse host remoto.
  - Se a única correspondência for um host remoto referenciado por um grupo de hosts, o aplicativo executará o operador nesse host remoto.
  - Se o aplicativo encontrar diversas correspondências, executará o operador no primeiro host remoto encontrado que seja referenciado por um touchpoint do proxy ou um grupo de hosts.
  - Se não houver correspondências para os critérios de pesquisa, continuará o processamento.

5. O aplicativo determina o motivo pelo qual o operador não foi executado. Em seguida, o aplicativo falha o operador com uma mensagem de falha.
  - **Condição:** nenhum dos identificadores recuperados na Etapa 1 pertence a algum host no ambiente atual.  
**Resultados:** o aplicativo falha o operador com a seguinte mensagem:  
A mensagem não pôde ser enviada ao nó.
  - **Condição:** um ou mais identificadores recuperados na Etapa 1 pertencem a um host onde todos os orquestradores e agentes estão inativos.  
**Resultados:** o operador falha com a seguinte mensagem:  
A mensagem não pôde ser enviada ao nó.
  - **Condição:** um ou mais identificadores da Etapa 1 pertencem a um host com todos os touchpoints desativados ou são referenciados apenas com touchpoints do proxy ou grupos de hosts desativados.  
**Resultados:** o aplicativo retorna um touchpoint ou grupo de hosts desativado no campo TouchpointName das variáveis de saída do sistema do conjunto de dados do operador. O aplicativo falha o operador com a seguinte mensagem de erro:  
O nó está desativado.

**Observação:** considere a exceção em que o usuário especifica um nome de host ou endereço IP no campo Destino de um operador *Chamar o Java*. O CA Process Automation procurará primeiro por um host do agente correspondente. Se tiver êxito, o operador Chamar o Java será executado no host do agente. Se a pesquisa falhar, o aplicativo procurará um host do orquestrador correspondente. Se for encontrado, ele tentará executar, mas o operador Chamar o Java falhará, pois não pode ser executado em um orquestrador.

**Mais informações:**

[Configurações de destino](#) (na página 231)

## Processando um destino especificado como a ID de um agente ou orquestrador

Suponha que você deseja que um operador use como destino o mesmo host sempre que for executado. Nesse caso, é possível especificar os dados do AgentID como um destino. Os dados exclusivos de AgentID são definidos durante a instalação de cada orquestrador e de cada agente. Em tempo de execução, o CA Process Automation recupera a ID do orquestrador ou do agente e a salva para o campo *AgentID* no conjunto de dados do sistema.

Após uma execução de teste de um operador, é possível copiar a sequência de caracteres da ID do agente exibida no conjunto de dados do sistema no campo *Destino*.

Quando você precisar que dois operadores consecutivos sejam executados no mesmo host do agente, especifique o mesmo AgentID como o destino dos dois operadores. É possível usar o AgentID como um destino alternativo para um determinado touchpoint.

**Importante:** o uso do AgentID se refere ao host local, no qual o agente ou o orquestrador está instalado. Quando você especifica o AgentID, o CA Process Automation não usa nenhum touchpoint do proxy ou grupo de hosts em execução no agente para a execução em hosts remotos. Considere o caso em que *Operator\_1* é executado em um touchpoint do proxy configurado em um agente com uma ID do agente especificada. Não use esse AgentID como o destino para o *Operator\_2*. Se o AgentID estiver especificado, o CA Process Automation não executará nesse touchpoint do proxy para o *Operator\_2*.

O CA Process Automation causa falhas no operador quando encontra qualquer uma das condições a seguir:

- O agente de destino ou o orquestrador está inativo. O CA Process Automation publica a seguinte mensagem:  
A mensagem não pôde ser enviada ao nó.
- Todos os touchpoints mapeados para o destino estão desativados. O CA Process Automation publica a mensagem a seguir com um dos touchpoints desativados:  
O nó está desativado.

**Observação:** o operador Chamar o Java só pode usar agentes como destino. O operador Chamar o Java não pode ser executado em um orquestrador. Uma ID no campo Destino do operador Chamar o Java deve ser resolvida para um agente (não um orquestrador), caso contrário, o operador falhará.

## Caso de Uso: rastrear processos recuperados por meio dos logs

O cenário a seguir é uma alternativa à abertura de vários processos a fim de identificar aqueles que foram recuperados automaticamente. Neste cenário, você examine os arquivos de log.

### Hipóteses:

- Um agente é mapeado para dois touchpoints.
- Mais de 1.000 processos estão em execução simultaneamente.
- A Recuperação automática de operadores está configurada para os touchpoints.

### Cenário:

1. O agente torna-se inativo de repente.
2. Enquanto o agente está inativo, 300 dos 1.000 processos tentam executar os operadores em dois touchpoints.
3. Os 300 processos entram no estado BLOCKED (Bloqueado).
4. O agente torna-se ativo.
5. A recuperação dos operadores, que é iniciada automaticamente, recupera os 300 processos.
6. Mais tarde, você observa que a execução dos 1.000 processos é concluída. Você abre um deles para examinar os logs e percebe que ele foi recuperado automaticamente.
7. Conclua uma das seguintes ações para saber quais dos 1.000 processos que concluíram a execução foram concluídos depois de serem recuperados:
  - Abra cada um dos 1.000 processos para determinar quais foram recuperados automaticamente.
  - Examine o arquivo de log.

**Observação:** o próximo tópico descreve a alternativa preferencial, examinando o arquivo de log.

## Exemplo da Recuperação automática dos Operadores

Se Recuperação automática de operadores (o padrão) estiver selecionada, não será necessário iniciar a recuperação. Cada orquestrador dentro do ambiente detecta que o agente ou orquestrador se torna ativo e executa a recuperação. Cada orquestrador executa a recuperação em qualquer touchpoint, grupo de hosts ou orquestrador ativado que esteja mapeado para o agente ou o orquestrador no ambiente.

Se um agente estiver mapeado para vários touchpoints, os orquestradores executarão a recuperação em todos os touchpoints ativados que estiverem mapeados para o agente. Quando um agente ou orquestrador se torna ativo, o orquestrador do domínio transmite as mudanças para os outros orquestradores. Em uma instalação agrupada, apenas o nó principal executa a recuperação.

### Exemplo: Recuperação automática

Considere o seguinte cenário, no qual um agente está mapeado para os seguintes touchpoints e grupos de hosts em um ambiente:

- TP\_user01-w500,
- TP\_user01-w500\_2
- TP\_user01-w500\_3
- TP\_manyAgents
- HG\_user01-w500

Suponha que todos os touchpoints e grupos de hosts estejam ativados e que somente os seguintes touchpoints e grupos de hosts estejam definidos para Recuperação automática de operadores:

- TP\_user01-w500
- TP\_user01-w500\_3
- HG\_user01-w500

Quando o agente se torna ativo, cada orquestrador no ambiente tenta executar a recuperação automática nos três touchpoints.

Suponha que existam dois orquestradores no ambiente:

**O arquivo de log Orchestrator1 contém:**

- Mensagem para sinalizar o agente que se tornou ativo novamente:

2010-06-28 19:22:59,984 DEBU,G

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Verificando se iniciará a recuperação do touchpoint/orquestrador dos operadores 'System\_Error' nos processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da recuperação automática configurado), executados com relação à ID do agente/servidor: f3492322-5517-4a21-8a19-92838ccb3f65

- O Orquestrador mapeou o agente para HG\_user01-w500:

2011-03-28 19:22:59,984 INFO

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando o touchpoint/orquestrador: HG\_user01-w500 Recuperação... Recuperando operadores 'System\_error' executados com relação a: HG\_user01-w500 e que pertencem aos processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da recuperação automática configurado).

2011-03-28 19:22:59,984 DEBUG

[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread] recoverSystemErrorProcesses: Recuperar os processos (com o sinalizador da recuperação automática ativado) nos estados 'Em espera' ou 'Em execução' ou 'Bloqueado' e que contenham operadores em 'System Error' executados com relação a: HG\_user01-w500

- O Orquestrador sinaliza 0 processo para ser recuperado para HG\_user01-w500:

2011-03-28 19:23:00,000 WARN

[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread] recoverSystemErrorProcesses: Recuperado - 0 processo para recuperar para a recuperação de HG\_user01-w500.

2011-03-28 19:23:00,000 INFO

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Nenhum Processo encontrado para recuperar para Touchpoint/Orquestrador: Recuperação de HG\_user01-w500.

- O Orquestrador mapeou o agente para TP\_user01-w500:

2011-03-28 19:23:00,000 INFO

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando o touchpoint/orquestrador: TP\_user01-w500 Recuperação... Recuperando operadores 'System\_error' executados com relação a: TP\_user01-w500 e que pertencem aos processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da recuperação automática configurado).

2011-03-28 19:23:00,000 DEBUG

[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread] recoverSystemErrorProcesses: Recuperar os processos (com o sinalizador da recuperação automática ativado) nos estados 'Em espera' ou 'Em execução' ou 'Bloqueado' e que contenham operadores em 'System Error' executados com relação a: TP\_user01-w500.

- O Orquestrador sinaliza 1 processo para ser recuperado para TP\_user01-w500:  
2011-03-28 19:23:00,015 WARN  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: Recuperado - 1 processos para recuperar para a recuperação de TP\_user01-w500.
- O Orquestrador coloca em fila a solicitação de recuperação para 1 processo (ROID: 2889) para ser recuperado para TP\_user01-w500:  
2011-03-28 19:23:00,015 WARN  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: Recuperação do enfileiramento de processo com ROID - 2889 para a recuperação de TP\_user01-w500.  
2011-03-28 19:23:00,015 DEBUG  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: resposta especial para a recuperação do processo. ROID é 2889 UUID é 3fe95f08-a347-4d6d-a0a3-b3639836e130  
2011-03-28 19:23:00,015 DEBUG  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: Resposta especial para a recuperação do processo foi publicada. ROID é 2889 UUID é 3fe95f08-a347-4d6d-a0a3-b3639836e130  
2011-03-28 19:23:00,015 INFO  
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Solicitações enfileiradas para Touchpoint: Recuperação de TP\_user01-w500... Recuperando processos com os seguintes ROIDs: 2889.
- O Orquestrador sinaliza que o touchpoint TP\_user01-w500\_2 está mapeado para o agente, mas não está configurado para recuperação automática:  
2011-03-28 19:23:00,015 DEBUG  
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Touchpoint: TP\_user01-w500\_2 mapeado para o agente: f3492322-5517-4a21-8a19-92838ccb3f65 não está configurado para recuperação automática dos operadores 'System\_Error' nos processos 'Em espera' ou 'Em execução' ou 'Bloqueado'.
- O Orquestrador mapeou o agente para TP\_user01-w500\_3:  
2011-03-28 19:23:00,015 INFO  
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando Touchpoint/Orquestrador: Recuperação de TP\_user01-w500\_3. Recuperando operadores 'System\_Error' executados com relação a: TP\_user01-w500\_3 e que pertençam aos processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da recuperação automática configurado).  
2011-03-28 19:23:00,015 DEBUG  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: Recuperar os processos (com o sinalizador da recuperação automática ativado) nos estados 'Em espera' ou 'Em execução' ou 'Bloqueado' e que contenham operadores em 'System Error' executados com relação a: TP\_user01-w500\_3

- O Orquestrador sinaliza 0 processo para ser recuperado para TP\_user01-w500\_3:  
2011-03-28 19:23:00,015 WARN  
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]  
recoverSystemErrorProcesses: Recuperado - 0 processo para recuperar para a  
recuperação de TP\_user01-w500\_3.  
2011-03-28 19:23:00,031 INFO  
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Nenhum Processo  
encontrado para recuperar para Touchpoint/Orquestrador: Recuperação de  
TP\_user01-w500\_3.
- O orquestrador sinaliza que o touchpoint TP\_manyAgents está mapeado para o  
agente, mas não está configurado para recuperação automática:  
2011-03-28 19:23:00,031 DEBUG  
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Touchpoint:  
TP\_manyAgents mapeado para o agente: f3492322-5517-4a21-8a19-92838ccb3f65 não  
está configurado para recuperação automática dos operadores 'System\_Error' nos  
processos 'Em espera' ou 'Em execução' ou 'Bloqueado'
- O Orquestrador captura a solicitação de recuperação na fila para o processo 2889:  
2011-03-28 19:23:30,015 INFO  
[com.optinuity.c2o.workflowengine.FlowManagerCache] [onPool Worker-3]  
Registrando gerenciador de fluxo: 2889  
2011-03-28 19:23:30,015 INFO  
[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3]  
Manipulando resposta: [\$\$TouchpointRecovery&&]. A identificação do fluxo é 2889.  
2011-03-28 19:23:30,171 INFO  
[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3] Iniciar  
recuperação do Processo com R0ID: 2889 para a recuperação de TP\_user01-w500.
- Orquestrador redefine o(s) Operador(es) com System\_Error que foram executados  
em relação a TP\_user01-w500 dentro do processo 2889:  
2011-03-28 19:23:30,171 INFO  
[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3]  
Redefinindo Operadores System\_Error no processo com R0ID: 2889 para a recuperação  
de TP\_user01-w500.  
2011-03-28 19:23:30,203 INFO  
[com.optinuity.c2o.workflowengine.workflowmanager] [onpool worker-3]  
Redefinindo Operador com R0ID: 2912 no Processo com R0ID: 2889 para a recuperação  
de TP\_user01-w500.

- O Orquestrador retoma todo o processo 2889 após a redefinição do Operadores apropriados:

2011-03-28 19:23:30,343 INFO

[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3] Retomando Processo com ROID: 2889 para a recuperação de TP\_user01-w500.

2011-03-28 19:23:30,343 INFO

[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3]

Reiniciando fluxo de trabalho - ROID é 2889.

2011-03-28 19:23:30,703 INFO

[com.optinuity.c2o.workflowengine.WorkflowManager] [onPool Worker-3] Conclusão da retomada do Processo com ROID: 2889 para a recuperação de TP\_user01-w500.

Processo 2889 concluído com êxito. Uma vez que o processo de recuperação foi iniciado automaticamente, ele registra o estado em que o mecanismo iniciou a recuperação. Para a recuperação manual, os logs incluem o nome do usuário que iniciou a recuperação.

**O arquivo de log Orchestrator2 (OrchestratorD610b) contém:**

- O seguinte para OrchestratorD610b é semelhante ao Orquestrador:

```
2011-03-28 19:23:41,444 INFO
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando
Touchpoint/Orquestrador: Recuperação de HG_user01-w500. Recuperando operadores
'System_Error' executados com relação a: HG_user01-w500 e que pertençam aos
processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da
recuperação automática configurado).
2011-03-28 19:23:41,444 WARN
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]
recoverSystemErrorProcesses: Recuperado - 0 processo para recuperar para a
recuperação de HG_user01-w500.
2011-03-28 19:23:41,444 INFO
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Nenhum Processo
encontrado para recuperar para Touchpoint/Orquestrador: Recuperação de
HG_user01-w500...
2011-03-28 19:23:41,444 INFO
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando o
touchpoint/orquestrador: TP_user01-w500 Recuperação... Recuperando operadores
'System_error' executados com relação a: TP_user01-w500 e que pertencem aos
processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da
recuperação automática configurado).
2011-03-28 19:23:41,444 WARN
[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]
recoverSystemErrorProcesses: Recuperado - 0 processo para recuperar para a
recuperação de TP_user01-w500.
2011-03-28 19:23:41,444 INFO
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Nenhum Processo
encontrado para recuperar para Touchpoint/Orquestrador: Recuperação de
TP_user01-w500...
2011-03-28 19:23:41,444 INFO
[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Iniciando o
touchpoint/orquestrador: TP_user01-w500_3 Recuperação... Recuperando operadores
'System_error' executados com relação a: TP_user01-w500_3 e que pertencem aos
processos 'Em espera' ou 'Em execução' ou 'Bloqueado' (com o sinalizador da
recuperação automática configurado).
```

- Orquestradord610b localiza 1 processo para recuperar para TP\_user01-w500\_3 (ROID do processo: 541) e coloca na fila a sua solicitação de recuperação:

2011-03-28 19:23:41,444 WARN

[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]

recoverSystemErrorProcesses: Recuperado - 1 processo a ser recuperado para a recuperação de TP\_user01-w500\_3.

2011-03-28 19:23:41,444 WARN

com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]

recoverSystemErrorProcesses: Recuperação em fila do processo com a recuperação ROID - 541 for TP\_user01-w500\_3.

2011-03-28 19:23:41,444 INFO

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Solicitações em fila para o touchpoint: TP\_user01-w500\_3 Recovery... Recuperando processos com os seguintes ROIDs: 541

2011-03-28 19:23:41,444 WARN

[com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]

recoverSystemErrorProcesses: Recuperado - 1 processo a ser recuperado para a recuperação de TP\_user01-w500\_3.

2011-03-28 19:23:41,444 WARN

com.optinuity.c2o.workflowengine.C2oWorkFlowEngine] [aListenerThread]

recoverSystemErrorProcesses: Recuperação em fila do processo com a recuperação ROID - 541 for TP\_user01-w500\_3.

2011-03-28 19:23:41,444 INFO

[com.optinuity.c2o.c2oserver.ServerManager] [aListenerThread] Solicitações em fila para o touchpoint: TP\_user01-w500\_3 Recovery... Recuperando processos com os seguintes ROIDs: 541

- O OrchestratorD610b captura a solicitação de recuperação, redefine os Operadores com System\_Error que foram executados com relação a TP\_user01-w500\_3 no processo e retoma o processo:

```
2011-03-28 19:24:11,461 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63]
Manipulando resposta: [${TouchpointRecovery&&}. A identificação do fluxo é 541.
2011-03-28 19:24:11,804 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63] Iniciar
Recuperação de processo com ROID: 541 para a recuperação de TP_user01-w500_3.
2011-03-28 19:24:11,820 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63]
Redefinindo Operadores System_Error no processo com ROID: 541 para a recuperação
de TP_user01-w500_3.
2011-03-28 19:24:11,883 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63]
Redefinindo Operador com ROID: 564 no Processo com ROID: 541 para a recuperação
de TP_user01-w500_3.
2011-03-28 19:24:12,039 INFO
[com.optinuity.c2o.workflowengine.C20SvcIconInstanceRefObject] [nPool
Worker-63] Inside Update ConnectorTraversed, Current/destination é
Start_System_Process_1 Source é Start_1
2011-03-28 19:24:12,117 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63] Retomando
Processo com ROID: 541 para a recuperação de TP_user01-w500_3.
2011-03-28 19:24:12,117 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63] Retomando
fluxo de trabalho - ROID é 541.
2011-03-28 19:24:12,133 INFO
[com.optinuity.c2o.workflowengine.C20SvcIconInstanceRefObject] [nPool
Worker-63] Inside Update ConnectorTraversed, Current/destination é
Start_System_Process_1 Source é Start_1
2011-03-28 19:24:12,648 INFO
[com.optinuity.c2o.workflowengine.WorkflowManager] [nPool Worker-63] Conclusão
da retomada do processo com ROID: 541 para a recuperação TP_user01-w500_3.
```